

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA STROJNÍHO INŽENÝRSTVÍ**  
**ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A**  
**ROBOTIKY**

FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

# SYSTÉMY PRŮMYSLOVÉHO VIDĚNÍ S ROBOTY KUKA A JEHO APLIKACE NA ROZPOZNÁVÁNÍ VOLNĚ LOŽENÝCH PRVKŮ

ROBOT VISION WITH INDUSTRIAL ROBOTS KUKA

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**BC. JAN KRUTÍLEK**

**VEDOUcí PRÁCE**  
SUPERVISOR

**ING. TOMÁŠ KUBELA**

BRNO 2010



# SEM BUDE VLOŽENO ORIGINÁLNÍ ZADÁNÍ DIPLOMOVÉ PRÁCE

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav výrobních strojů, systémů a robotiky

Akademický rok: 2009/2010

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

student(ka): Bc. Jan Krutílek

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Výrobní stroje, systémy a roboty (2301T041)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

**Systémy průmyslového vidění s roboty Kuka a jeho aplikace na rozpoznávání  
volně ložených prvků**

v anglickém jazyce:

**Robot vision with industrial robots Kuka**

Stručná charakteristika problematiky úkolu:

Úkolem je seznámit se s možnostmi strojového vidění u průmyslových robotů Kuka a vytvořit prakticky orientovanou demonstrační aplikaci.

Cíle diplomové práce:

- (1) Seznamte se s principy robotického vidění v průmyslových aplikacích, jeho výhodami a nevýhodami. Vyhotovte také rešerši možných řešení, které jsou v současné době dostupné na trhu.
- (2) Seznamte se s kamerovým systémem Siemens VS722, jeho možnostmi a programováním a možnými způsoby komunikace s průmyslovými roboty Kuka.
- (3) Vytvořte demonstrační aplikaci na průmyslovém robotu Kuka KR 3. Zaměřte se především na přesné rozpoznání libovolně umístěných prvků na ploše a s tím související přesné navedení robotu za účelem manipulace nalezených prvků.

Seznam odborné literatury:

- [1] L. Sciavicco, B. Siciliano: Robotics Modelling, Planning and Control. Springer London, 2008
- [2] PIRES, J. N. Industrial Robots Programming: Building Applications for the Factories of the Future. Springer, 2007. 282 s. ISBN 978-0-387-23325-3
- [3] WOLF, A., STEINMANN, R. SCHUNK, H. Grippers in Motion: The Fascination of Automated Handling Tasks. Springer, 2005. 242 s. ISBN 978-3-540-27718-7

Vedoucí diplomové práce: Ing. Tomáš Kubela

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2009/2010.

V Brně, dne 26.11.2009

L.S.

---

Ing. Petr Blecha, Ph.D.  
Ředitel ústavu

---

doc. RNDr. Miroslav Doupovec, CSc.  
Děkan fakulty

## **Abstrakt**

Bc. Jan Krutílek

Robotické vidění s průmyslovými roboty Kuka

Diplomová práce, Ústav Výrobních Strojů, Systémů a Robotiky, 2010.

Diplomová práce pojednává o robotickém vidění a jeho aplikaci na problém manipulace volně ložených prvků. Je zde uveden přehled principů dnes nejpožívanějších kamerových systémů. S ohledem na řešení dané aplikace jsou uvedeny možnosti využití základních softsenzorů při rozpoznávání různých objektů. Úkolem práce je také naprogramování a realizace demonstrační úlohy při využití znalostí programování PLC, znalostí expertního programování robota v jazyce KRL, znalostí psaní skriptů pro inteligentní kameru v programu Spectation a využití znalostí síťové komunikace mezi použitými zařízeními.

Klíčová slova: strojové vidění, inteligentní kamera, Kuka robot, rozpoznávání objektů

## **Abstract**

Bc. Jan Krutílek

Robotic vision with industrial robots Kuka

Master's thesis, Institute of Production Machines, Systems and Robotics, 2010.

Diploma thesis deals with a robot vision and its application to the problem of manipulation of coincidentally placed objects. There is mentioned an overview of current principles of the most frequently used vision systems on the market. With regard to the required task to be solved, there are mentioned various possibilities of using basic softsensors during the recognition of different objects. The objective of this Diploma thesis is also programming and realization of a demonstration application applying knowledge of PLC programming, knowledge of expert programming KRL language (for KUKA robots), knowledge of designing scripts for smart camera in Spectation software and knowledge of network communication among all devices used in this case.

Key words: machine vision, smart camera, Kuka robot, object recognition

**Bibliografická citace:**

KRUTÍLEK, J. *Systémy průmyslového vidění s roboty Kuka a jeho aplikace na rozpoznávání volně ložených prvků*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 60 s. Vedoucí diplomové práce Ing. Tomáš Kubela.

## **Čestné prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, pod vedením ing. Tomáše Kubely a s využitím zdrojů uvedených v seznamu literatury.

V Brně, dne 27.5. 2010

.....

Bc. Jan Krutílek

## **Poděkování**

Na tomto místě bych chtěl poděkovat především ing. Tomáši Kubelovi za jeho čas a cenné rady při řešení a tvorbě této diplomové práce.



## Obsah

1 Úvod.....	10
2 Kamerate systémy.....	11
2.1 PC systémy.....	11
2.2 Kompaktní systémy - CVS (Compact Vision System) .....	12
2.3 Kamerate senzory.....	12
2.4 Inteligentní kamery (smart camera) .....	13
2.5 Principy robotického vidění.....	15
2.5.1 2D vidění.....	15
2.5.2 3D vidění.....	15
2.5.3 Osvětlení.....	16
2.6 Průzkum trhu v oblasti použité kamery .....	17
2.6.1 Inteligentní kamery.....	17
2.6.2 Ostatní kamery.....	20
2.7 Možnosti sítí .....	21
2.7.1 CAN .....	21
2.7.2 DeviceNet .....	21
2.8 Uživatelské prostředí SIMATIC Spectation 2.8.2 .....	22
2.8.1 Foreground skript.....	23
2.8.2 Background skript .....	24
2.8.3 Kalibrace kamery .....	24
3 Koncepce pracoviště .....	26
3.1 Schéma .....	26
3.2 Robot.....	26
3.3 PLC .....	27
3.4 Kamera .....	27
3.5 Popis komunikace .....	28
4 Popis a návržení koncepce řešení .....	31
4.1 Popis demonstrační úlohy .....	31
4.2 Rozbor problematiky manipulace volně ložených prvků .....	31
4.2.1 Koncepce úlohy podle pohyblivosti objektu.....	32
4.2.2 Problematika rozpoznávání objektu .....	32
4.2.3 Problematika uchopení .....	37
4.2.4 Vzdálenost kamery od pracovní plochy.....	38
5 Demonstrační úloha .....	39
5.1 Inspekce .....	41
5.2 Popis komunikace server a klient .....	41
5.3 Popis činnosti PLC .....	45
5.4 Popis činnosti robotu .....	48
6 Závěr.....	50
Použitá Literatura .....	51
Seznam použitých zkratk .....	54
Seznam příloh .....	55

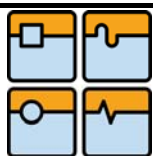
## 1 Úvod

V poslední době dochází, nejvíce ve velkosériové výrobě, ke snaze automatizovat průmyslovou výrobu na různých úrovních. Především odstraňovat stereotypní, nebezpečnou nebo namáhavou práci ve výrobě, montáži, kontrole a paletizaci produktů. Řešením, které se přímo nabízí je využití průmyslových robotů a manipulátorů. Tyto stroje navíc často zvyšují efektivnost a produktivitu výroby a tím potažmo snižují celkové výrobní náklady připadající na jeden kus produktu.

Nasazení kamer v průmyslu není dnes již také žádnou novinkou. S prudkým vývojem informačních technologií se ale velice změnila možnost jejich využití. Dnes v této oblasti stále se rozšiřující skupinou jsou kamery vybavené vlastním řídicím a vyhodnocovacím systémem. Tyto kamery se používají buď samostatně, nebo jako rovnocenné doplňující zařízení schopné komunikovat a předávat potřebné informace dalším strojům ve výrobě.

Spojení kamerových a robotických systémů, tak jako nasazení jiných externích snímačů robotu dříve, přináší značné rozšíření aplikací PRaM v průmyslové praxi. Je to způsob, jak může robot získat informace například o přítomnosti součásti v daném prostoru, a její případné poloze, tvaru, rozměru, natočení, barvě součásti nebo i informaci o některých parametrech jejího pohybu a tyto informace použít k následné manipulaci se součástmi. Z toho vyplývají možnosti uplatnění v různých odvětvích průmyslu. Strojové vidění navíc vede ke zvýšení pružnosti výroby a zvýšení autonomnosti chování. Proto mají tyto systémy velký potenciál a můžeme předpokládat, že se budou dále vyvíjet a rozšiřovat.

Cílem této práce je mimo jiné vytvoření demonstrační aplikace strojového vidění, což vyžaduje seznámení se s kamerovým i robotickým systémem a jejich součinnostmi na jednom pracovišti strojového vidění. V rámci demonstrační aplikace je tato práce zaměřena na rozpoznání a manipulaci prvků, volně ložených v daném pracovním prostoru.



## 2 Kamerové systémy

Vzhledem k technickému pokroku, nejvýraznějším ve stupni miniaturizace a zvyšování výkonových parametrů dochází k stále většímu nasazování kamerových systémů v oblasti průmyslu i v oblasti zabezpečování.

V oblasti průmyslových aplikací se v současné době používá několik druhů kamerových systémů:

### 2.1 PC systémy

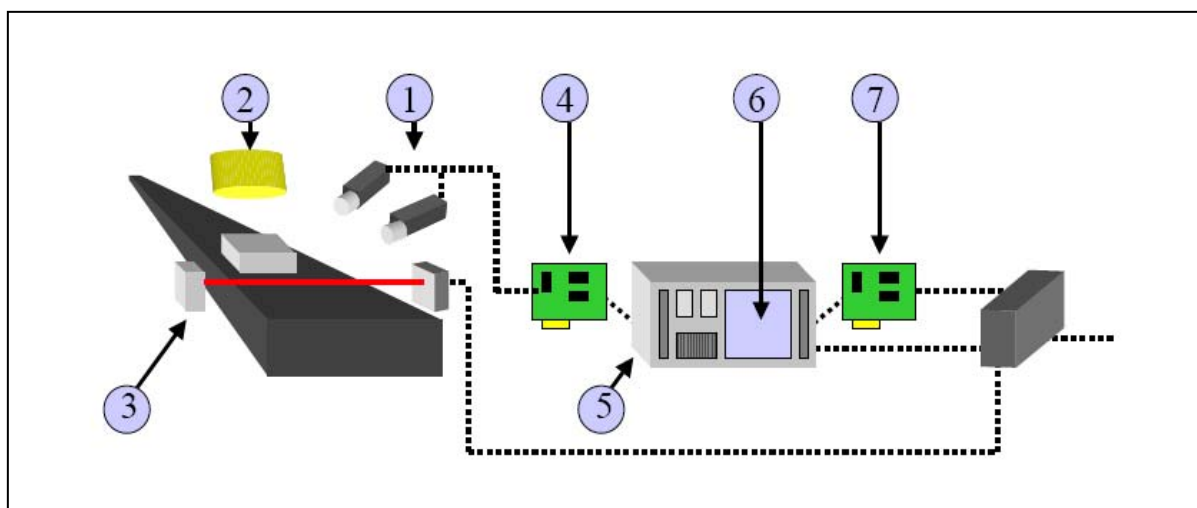
U těchto systémů je počítač ústředním prvkem. Kamera snímá scénu a pořizuje snímky a posílá je po síti do PC.

Pokud má kamera analogový výstup, tak je před zpracováním v počítači nejprve potřeba použít grabovací karty (tzv. frame grabber). Frame grabber je zařízení, které tyto snímky digitalizuje pro zpracování v počítači.

Pokud má kamera digitální výstup, tak snímky posílá rovnou do PC přes standardní rozhraní - Ethernet, USB nebo Firewire.

Následně probíhá v inspekčním softwaru vyhodnocení snímku (kontrola úplnosti součástí, atd.) a na základě těchto výsledků počítač nařídí případný akční zásah. Většinou tyto systémy bývají doplněny externími snímači (např. optická závora), které mohou zapínat spoušť kamery nebo řídit některé další funkce systému.

Tyto systémy se využívají především pro možnost připojit více kamer nebo v neprůmyslové oblasti při pokročilých metodách počítačového vidění [1].



1 - kamery a optika

2 - osvětlení

3 - vnější snímač

4 - frame grabber

5 - PC

6 - software pro inspekci

7 - digitální I/O a připojení k síti

obr. 2.1 Hlavní komponenty "PC systémů" [1]

## 2.2 Kompaktní systémy - CVS (Compact Vision System)

V podstatě se jedná o PC systémy, u kterých je PC upravené pro konkrétní aplikaci. Na rozdíl od PC systémů tyto systémy nemívají pevný disk (takže jsou odolnější vůči vibracím). Kompaktní kamerové systémy jsou určeny, mimo jiné, pro prostředí s extrémními podmínkami jako je prašnost, teplota, vibrace a znečištění. S ohledem na tyto vlivy je přizpůsobena konstrukce kamery - kamera bývá v neperforovaném odolném pouzdru. Předávání dat řídicí jednotce probíhá pomocí sběrnice FireWire. Také má VGA výstup, takže lze po připojení obrazovky sledovat v reálném čase činnost kamery a jejích inspekci. S okolím (PLC aj.) komunikuje prostřednictvím digitálních I/O. Pro ukládání programů je možno použít paměť flash a RAM. Systém umožňuje pomocí sběrnice FireWire připojení jedné nebo i více kamer [2].



obr. 2.2 Kompaktní kamerový systém firmy National Instruments [2]

## 2.3 Kamerové senzory

U těchto systémů jsou osvětlení i samotný kamerový senzor integrovány do jednoho celku, doplněno řídicí jednotkou s kterou komunikuje přes Ethernet nebo Profibus a kterou můžeme zvolit dle našich potřeb. Zvolená řídicí jednotka hlavní měrou ovlivní mezní složitost aplikace pro kterou je kamerový senzor využit. Systém má menší počet digitálních I/O a malý počet nástrojů pro inspekci (u systému ZFV firmy Omron je to například 8 základních nástrojů [3]), z nichž se vybere 1 pro aktuální inspekci (většinou tedy nelze nástroje spolu kombinovat). Vzhledem k omezeným možnostem programování je často digitálním I/O přiřazena konkrétní funkce (synchronizace pořízení snímku atd.).

Kamerové senzory tvoří základní elementy strojového vidění. Obecně se hodí spíše pro méně náročné aplikace, kde je hlavní těžiště jejich využití, protože jsou jednoduché a cenově dostupné. Jsou vybaveny jen omezeným množstvím funkcí a algoritmů. V případě složitějších aplikací je ale lze seskupovat tak aby každý senzor



prováděl část úlohy a spolu obhospodařovaly celou inspekci. Příkladem aplikací může být kontrola přítomnosti uzávěru či kontrola etiket [5] [3].



obr. 2.3 Kamerový senzor firmy Omron [4]

## 2.4 Inteligentní kamery (smart camera)

Pojmem inteligentní kamera (dále jen IK) se myslí zařízení pro strojové vidění v kompaktním pouzdru, které v sobě zahrnuje kameru doplněnou o řídicí, vyhodnocovací a komunikační jednotku a které je schopno provádět i poměrně náročné komplexní inspekce.

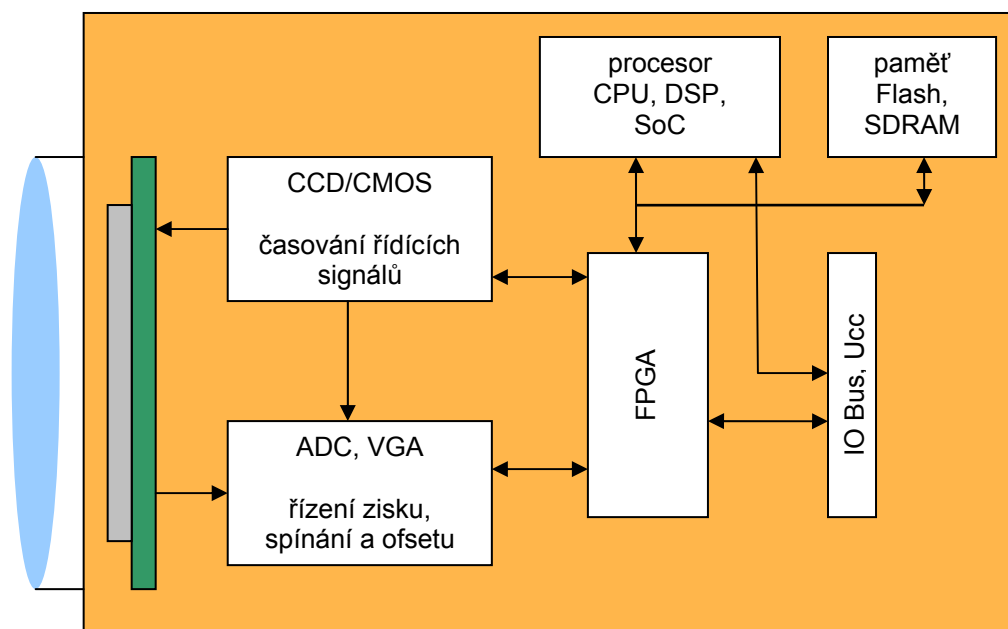
Kamera sejme obrázek ve formě jedno nebo více rozměrného signálu, který bude dále zpracováván. Kamera je za tímto účelem vybavena výkonným mikropočítačem případně i digitálním signálovým procesorem (DSP). Za účelem nahrávání a spouštění inspekčních programů je zde paměť flash a RAM. K urychlení některých standardních operací potřebných při analýze obrazu (filtrace, hledání hran aj.) se někdy využívají i velmi rychlé automaty programované v hradlových polích [6] [7].

K dnes už běžným vlastnostem IK patří řízení expozice prostřednictvím elektronické závěrky, přesné zahájení snímání, řízení světla při expozici. Využití IK roste hlavně díky stále se zvyšující výkonnosti kamer a jejich schopnosti spolupracovat v síti.

Komunikačním rozhraním je většinou Ethernet (kvůli požadavku rychle přenášet data) a slouží k připojení zařízení HMI (Human-Machine-Interface) (například PC), komunikaci s nadřazeným systémem nebo k připojení modulu pro rozšíření vstupů a výstupů.

Kromě Ethernetu bývají IK vybaveny ještě pomocným sériovým rozhraním RS-232/422/485 kvůli snadnějšímu připojení (většiny) PLC [7] [8].

Často se v průmyslu také používá také sběrnice CAN, která je robustní, odolná proti rušení a umožňuje přenášet obrazové informace v dostatečné kvalitě i kvantitě [6] [7] [8].



obr. 2.4 Schéma inteligentní kamery [8]

ADC - Analog to Digital Converter - digitálně analogový převodník

CCD - Charge-Coupled Device - polovodičový snímací prvek

CMOS - Complementary Metal-Oxide Semiconductor - polovodičový snímací prvek

CPU - Central Processor Unit - procesorová jednotka

DSP - Digital Signal Processor - procesor, který slouží k periodickému zpracování velkých objemů dat

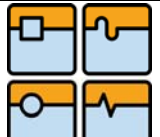
FPGA - Field Programmable Gate Array - obvod starající se o akvizici (vyzískání) dat ze snímacího prvku

I/O Bus - vstupně výstupní sběrnice

SoC - System on Chip - obvody, u nichž je snaha integrovat všechna potřebná zařízení pro činnost kamery na jediný čip

Ucc - napájecí napětí

VGA - standard pro počítačovou zobrazovací techniku

	Ústav výrobních strojů, systémů a robotiky	Str. 15
	DIPLOMOVÁ PRÁCE	

## 2.5 Principy robotického vidění

Průmyslové vidění se stejně jako jiné obory dále dělí na obory užší, které vycházejí z cílů a účelů, kterých je potřeba dosáhnout při minimálním vynaloženém úsilí i minimálních vynaložených finančních prostředcích.

Proto tyto řešení, lišící se v použité technologii, mnohdy nejsou navzájem svými konkurenty. Každá z těchto technologií má svůj cíl a proto i vývoj v těchto technologiích se může ubírat různým směrem, ale vždy k naplnění cílů, které však mohou být odlišné.

Základní způsoby rozponávání objektů v průmyslovém vidění jsou 2D rozpoznávání objektů a 3D rozpoznávání objektů [31].

### 2.5.1 2D vidění

Dvourozměrné rozpoznávání objektů se používá v řadě aplikací (při kontrole kvality, určování polohy atd.). Technologicky jde o srovnávání obrazu s naučeným vzorem, nebo hledání naučeného vzoru ve snímku získaném kamerou. Přičemž informace o tvaru součásti jsou získávány na základě intenzity jednotlivých částí obrazu [9].

U 2D rozpoznávání objektů je důležité zajistit aby se kontrolovaný prvek vyskytoval na scéně ve zhruba stejné poloze (např. čtení štítků na lahvích jedoucích po dopravníku). A také je nutné zajistit pokud možno neměnné osvětlení snímané scény, což se provádí většinou připevněním speciálního osvětlovače přímo na objektiv kamery nebo použitím přídavného externího zdroje světla.

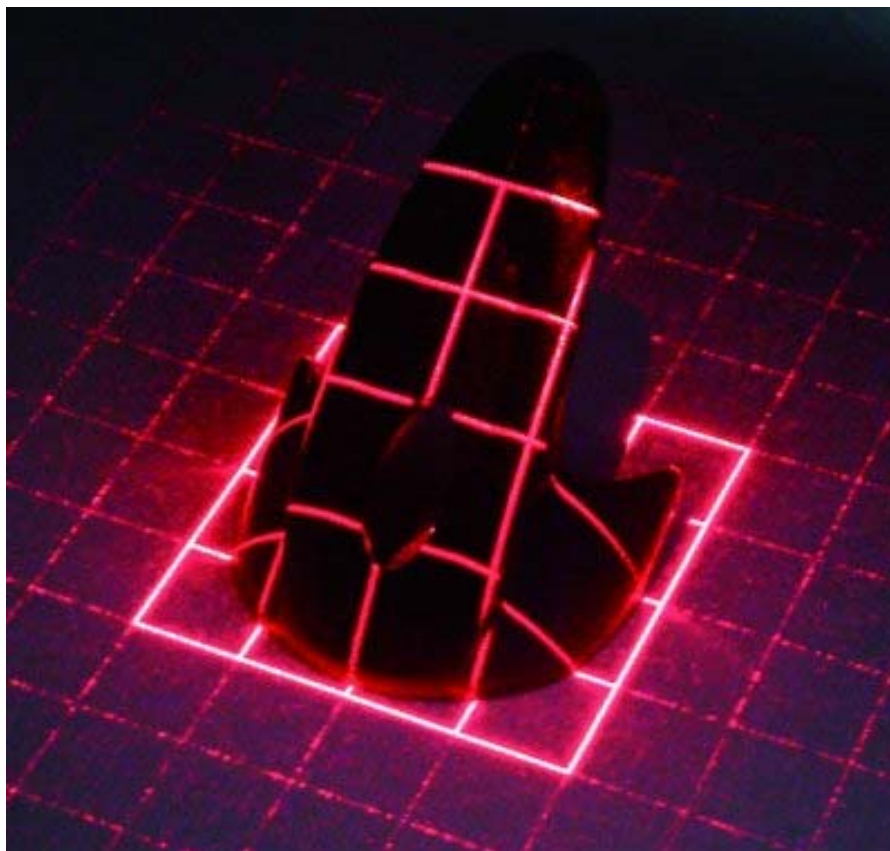
### 2.5.2 3D vidění

Systémů pro trojrozměrné rozpoznávání objektů je více. Primárně je rozdělujeme na pasivní (příkladem jsou systémy, pracující na principu stereovize) a aktivní systémy (příkladem jsou systémy, pracující na principu laserové projekce systémů).

V systémech pracujících na principu stereovize jsou informace o žádaném objektu získávány pomocí jedné nebo i více kalibrovaných kamer. Kamera pořídí snímek objektu z různých směrů. Pomocí sofistikovaných srovnávacích algoritmů určí odpovídající si body v různých snímcích a z nich sestaví model objektu. Tímto způsobem je však z obrazu získáváno menší množství informací o objektu a je proto nutné těchto snímků pořídít co nejvíce. Systémy pracující na tomto principu bývají označovány jako pasivní systémy [10][11].

Systémů pracujících na principu laserové projekce je více. Jeden ze základních principů je princip triangulace. Kamery pracující na tomto principu pracují tak, že projektor (většinou integrován do těla kamery) osvětluje předmět rovnoběžnými světelnými pruhy a kamera následně zachycuje výsledný vzor a vytváří výškový profil. Počítačový program následně vytvoří trojrozměrný obraz předmětu. Systémy pracující na tomto principu bývají označovány jako aktivní systémy (viz obr. 2.5) [9] [11] [12].





*obr. 2.5 Aktivní kamerové systémy [12]*

### 2.5.3 Osvětlení

Jedním z faktorů, které nejvíce ovlivňují kvalitu i dobu zpracování obrazu je osvětlení. Při špatném osvětlení vzniká zašuměný obraz (jakoby šedé tečky na pozadí při snímání jednoduše černého obrazu). Tento šum potom komplikuje rozpoznávání hran, měření přesného rozměru a plochy objektu a nutí nás k předzpracování obrazu nebo jiným opatřením. Problém je taky nekonstantní osvětlení, které komplikuje rozpoznávání naučených objektů. Kvůli kompenzaci nedostatečného množství světla nebo nekonstantních podmínek osvětlení se používají externí osvětlovače [31][32][33].

Typů osvětlovacích komponent pro strojové vidění je celá řada. Osvětlovače můžeme dělit podle vhodnosti pro různá prostředí i konkrétní aplikaci podle různých hledisek (např. světelný výkon, tvar osvětlovače, barva světla, aj.). Z hlediska tvaru jsou často používány osvětlovače kruhové a plošná osvětlovací pole. Z hlediska barvy světla se nejčastěji používají osvětlovače s bílým nebo červeným světlem. Tato problematika je podrobněji rozebrána v [13].

Pro osvětlení součástí, které obsahují poměrně malé vystouplé detaily (vyražený text, atd.) se používají zdroje difuzního světla. Toto světlo je rozptýlené a nevrhá tolik stínů, takže jsou tyto vystouplé detaily lépe rozpoznatelné.



## 2.6 Průzkum trhu v oblasti použité kamery

Ze zadání plyne požadavek na vyhotovení přehledu kamer (a částečně i jejich porovnání vzhledem k potřebám pro moji demonstrační aplikaci), které jsou v současné době (březen, 2010) dostupné na trhu.

V současné době je na trhu dostupné velké množství snímacích senzorů a kamer. Také neustále roste množství firem, které dodávají základní komponenty i příslušenství pro oblast strojového vidění. Proto ze všech variant vyberu jen několik zástupců. Všechny hodnocené kamery byly vybrány na základě požadavků a minimálních potřeb mojí úlohy. Vybrané kamery tedy snímají v odstínech šedé a s rozlišením kolem 640x480 pixelů a mají distributory v České republice.

Kapitola je dělena na část inteligentní kamery a část ostatní kamery, které většinou nemají vlastní procesor a ke své činnosti potřebují PC na kterém běží inspekční prostředí (většinou založené na bázi jazyka C nebo C++ ).

### 2.6.1 Inteligentní kamery

#### IMPACT A20 firmy PPT Vision

Firma PPT Vision je jedna z předních amerických firem, která navrhuje, vyrábí a prodává kamery pro strojové vidění. Výhradním dodavatelem této firmy v České republice je firma FCC Průmyslové Systémy, a.s. Z poměrně širokého sortimentu produktů této firmy jsem vybral inteligentní kameru IMPACT A20.

Tento systém má několik výhod. První je možnost použití testovacího softwaru (emulátor i plná verze softwaru je dodána při koupi kamery). Druhou je to, že kamera je vybavena flash pamětí o kapacitě 512MB. Jako další výhodu vidím poměrně velké možnosti sítí - při standardním provedení je kamera vybavena pro komunikaci po sériové lince, Ethernetu a Modbusu (další je možno dokoupit) a digitálními I/O. Jako nevýhodu vidím to, že kamera nepodporuje přímou spolupráci s robotem a také poměrně vysokou cenu [14][15].

#### SIMATIC VS722A firmy SIEMENS

Firma SIEMENS je původem německá firma, která v současné době zasahuje do mnoha odvětví průmyslu po celém světě. Výrobky firmy SIEMENS v České republice distribuuje její pobočka, firma Siemens, s.r.o. Z produktů této firmy jsem vybral inteligentní kameru SIMATIC VS722A.

Hlavními rysy této kamery je 1/3 palcový CCD snímač s rychlostí snímání 60 snímků za vteřinu při rozlišení 640x480 bodů. Kamera komunikuje s okolím pomocí Ethernetu a protokolu TCP/IP a dalších (viz. kapitola 4.4) a digitálních vstupů a výstupů.

Firma Siemens dává k dispozici plnou verzi vývojového softwaru, který obsahuje i emulátor pro ladění inspekčních programů. Tato kamera má poměrně univerzální použití [5][16].

Nevýhodou je to, že i když kamera podporuje přímou komunikaci s řídicími systémy řady robotů, bohužel roboty firmy KUKA mezi ně nepatří.

### IVC-2DM1111 firmy SICK

Firma SICK byla založena v roce 1946 v Německu a dnes patří k předním mezinárodním firmám dodávající jednotlivé komponenty i celé systémy pro oblast senzorických systémů [12]. Co se týká kamerových systémů, tak má tato firma v nabídce řadu inteligentních kamer pro 2D i 3D strojové vidění. V České republice je tato firma zastoupena firmou SICK spol. s.r.o. .

Tato kamera je vybavena CCD snímačem se standardním rozlišením 640x480 bodů. Dále je pak vybavena 800MHz procesorem a 128MB paměti a 16MB paměti flash, řadu digitálních I/O a podporovaná rozhraní Ethernet a RS485.

Nevýhodou této kamery je poměrně velká hmotnost, velké vnější rozměry a vysoká cena.

Vývojovým prostředím kamer SICK je IVC studio. Tento software je společný pro 2D i 3D kamery a má řadu propracovaných nástrojů, které zjednodušují programování kamery i při složitějších inspekcích. Příkladem může být nástroj, který má v sobě integrovaný nástroj pro čtení znaků (OCR) a nástroj pro vyhledání polohy a natočení. Software je včetně emulátoru zdarma při koupi kamery.

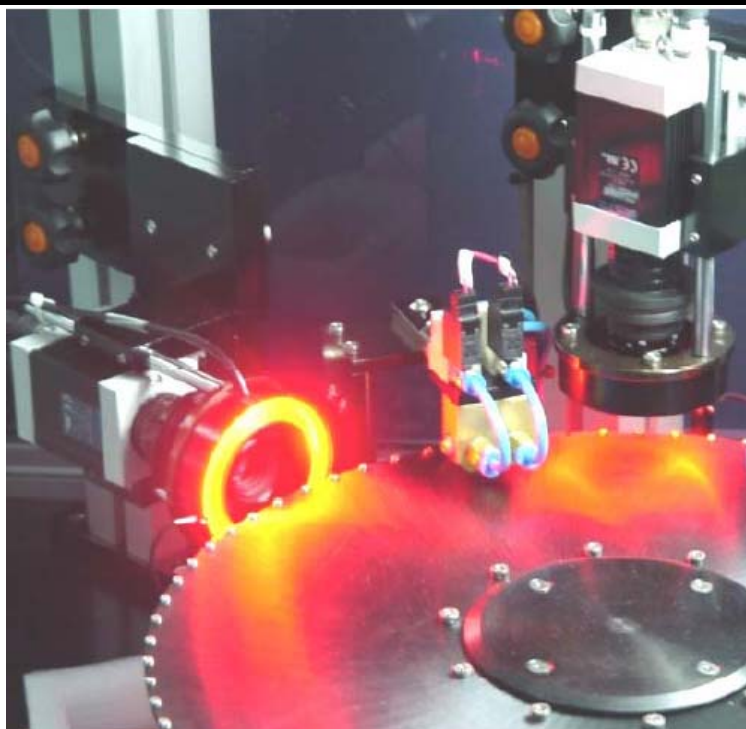
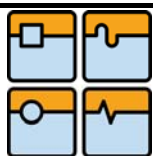
### SBOC-Q firmy FESTO

Firma FESTO je velice rozšířená firma v oblasti automatizace a v České Republice má svou pobočku Festo, s.r.o. Z jejích výrobků jsem vybral kameru SBOC-Q. Podrobnější informace jsou například v [17][18].

Tato kompaktní kamera je určena ke kontrole kvality a v základním provedení je vybavena konektory pro komunikaci po Ethernetu (přes protokoly TCP/IP, TelNet a další) a CanBusu. Ke kameře je software, který obsahuje i offline parametrizaci inspekčních produktů na obrázcích z kamery (emulátor). Kamera umožňuje přesné polohování elektrických pohonů přímo ze softwaru kamery.

Nevýhodou je, že tento software není volně dostupný, ale musí se koupit a také je nevýhodou poměrně vysoká pořizovací cena.

Výhodou je možnost snímání až 150 snímků za vteřinu a to, že kamera podporuje spolupráci s roboty.



obr. 2.6 Aplikace dvou kamer SBOC-Q firmy FESTO [18]

### IN-Sight 5400 firmy COGNEX

Firma COGNEX je významná, celosvětově rozšířená firma, která se zabývá kamerovými systémy [19]. Výrobky této firmy dodává do ČR řada firem, například firma Bartech, s.r.o. .


Vybraná kamera je monochromatická s rozlišením 640x480 bodů a má rychlost snímání 60 snímků za vteřinu. Software pro programování této kamery je v omezené 30 denní verzi je včetně emulátoru pro vyzkoušení volně ke stažení. Jmenuje In-Sight explorer a v plné verzi obsahuje dvě modulace. První - EasyBuilder - je pro rychlé programování s omezenou možností nastavení některých speciálních parametrů. Druhá - SpreadSheet - je určena pro náročnější inspekce a umožňuje detailnější nastavení a více parametrů a proto klade větší nároky na programovací znalosti a znalosti tohoto softwaru. Software této firmy patří k jednomu z nejpropracovanějších softwarů v této oblasti.

Kamera je vybavena nástrojem PatMax. Tento nástroj se používá při učení tvaru objektů. Pracuje tak, že porovnávání rozdílů šedé barvy převádí se na vektory - tedy neprobíhá na základě hodnoty intenzity pixelu, což způsobuje při rozpoznávání vyšší odolnost naučeného objektu proti nekonstantním podmínkám osvětlení.

K případné vizuální kontrole (např. zobrazení snímku a výsledků inspekce, kdy součást nevyhovuje) se používá dotykový panel uzpůsobený pro prostředí průmyslu.

Software také podporuje spolupráci s roboty a navíc, pokud je celý stroj řízen systémem, tak lze přímo v softwaru stroje používat výsledky z kamery.

Kamera komunikuje přes Ethernet nebo RS232 a 2 digitálních vstupů a 1 digitálního výstupu (modul s přídatnými digitálními I/O lze dokoupit).

	Ústav výrobních strojů, systémů a robotiky	Str. 20
	DIPLOMOVÁ PRÁCE	

## 2.6.2 Ostatní kamery

I tady v oblasti kamer založených na principu PC systémů je poměrně široký výběr produktů i firem. Vybral jsem jen několik málo z tohoto množství.

### Basler A601f firmy Basler

Tato německá firma byla založena v roce 1988 a dnes ji lze zařadit mezi světově rozšířené výrobce digitálních kamer pro různá odvětví. Distributorem této firmy v České republice je firma ELCOM, a.s. Vzhledem k požadavkům mé demonstrační aplikace jsem vybral kameru BASLER A601f.

Kamery firmy Basler jsou v základním provedení vybaveny jedním komunikačním prostředím. V případě této kamery je to FireWire rozhraní, což je běžné rozhraní, vhodné i pro aplikace v průmyslu. Ke snímání je použit 1/2 palcový prvek typu CMOS. Dále je důležité, že veškeré zpracování dat (obrázků) probíhá v externím počítači, na kterém běží programy pro inspekci a komunikaci [20].

Výhodou této kamery je nízká cena. Nicméně, vzhledem k možnostem kamery, si myslím, že je vhodná spíše pro zákazníky s velmi jasnou představou o potřebách konkrétní aplikace a požadavcích na poptávaný systém strojového vidění.

### CV-A10CL firmy JAI-PULNiX

Společnost JAI (formálně JAI PULNiX) je severoamerická odnož společnosti JAI a.s., celosvětově působícího výrobce CCD i CMOS kamer pro oblast strojového vidění i pro jiné oblasti použití. Distributorem této firmy v České republice je firma ELCOM, a.s. Z nabídky této firmy jsem vybral kameru CV-A10CL.

Snímání této kamery je zajištěno prostřednictvím 1/2 palcového CCD snímače, který může snímat rychlostí 60 snímků za vteřinu. Pro komunikaci s okolím využívá tato kamera rozhraní Camera Link nebo RS-232. Více informací je v [21].

Při průzkumu možností této kamery jsem narazil na několik nevýhod. Jednak to jsou už výše zmíněné, podle mne celkem omezené možnosti síťových rozhraní. Dále pak potřeba účelových komponent, které je třeba dokoupit a které zvyšují - byť nízkou - základní pořizovací cenu. Jedná se především o speciální frame grabber, který není součástí základního provedení.

Závěrem této kapitoly mohu říct, že nabídka kamer různých výrobců je velmi široká a pro danou úlohu určitě není jen jediná správná kamera. Vzhledem k vybavení laboratoří ÚVSSR bude diplomová práce realizována na kameře SIMATIC VS722A firmy SIEMENS. Tato kamera velkou mírou splňuje požadavky vyplývající z potřeb méj demonstrační úlohy.

## 2.7 Možnosti sítí

### 2.7.1 CAN

Controller Area Network (CAN) je rychlá sériová sběrnice, která byla navržena aby poskytla účinnou, spolehlivou a velmi ekonomickou vazbu mezi mikropočítači, různými inteligentními senzory a akčními členy. CAN využívá kroucenou dvoulinku ke komunikaci rychlostí až 1 Mbit/s s jedním až čtyřiceti zařízeními. Původně byla tato sběrnice vyvinuta pro zjednodušení elektrorozvodu v automobilech. Vzhledem k snadné rozšiřitelnosti, vysoké spolehlivosti a odolnosti při extrémních podmínkách (rušení, teplota, vibrace) se CAN rozšířil i do dalších průmyslových oblastí [23].

CAN umožňuje standardizovat komunikační objekty pro zpracování dat, datové služby, správu sítě, synchronizaci a nouzová hlášení. Je základem několika sběrnic, například DeviceNet.

Sběrnice CAN je navržena aby umožňovala distribuované řízení systémů v reálném čase. Protokol CAN je typu multi-master, což znamená, že každý uzel v síti může být master a řídit tak chování jiných uzlů. Tento typ řízení zvyšuje spolehlivost a zjednodušuje řízení. V případě, že dojde k poruše jednoho místa (uzlu) v síti, celé síti se nesníží funkčnost, ale přenesení se jen zpráva s nejvyšší prioritou. Řízení přístupu k médiu pracuje na principu náhodného přidělování, které kolize řeší na základě prioritního rozhodování. Zprávy vysílané po sběrnici neobsahují žádné údaje o tom, komu jsou určeny a jsou přijímány všemi uzly na sběrnici. Každá zpráva má identifikátor, podle kterého každý uzel sítě pozná, zda je zpráva adresovaná pro něj a také podle identifikátoru určí prioritu zprávy a její význam. Každý uzel je vybaven tzv. Acceptance filteringem, jehož úkolem je to aby uzel přijímal a dále zpracovával jen zprávy adresované právě jemu [22] [34].

### 2.7.2 DeviceNet

Jedná se o jeden z nejrozšířenějších protokolů založených na bázi CAN, který je upraven pro účely výrobních sítí. DeviceNet je jednoduchým způsobem řešený standardní otevřený protokol, který umožňuje přenos dat na nejnižší úrovni řízení průmyslových zařízení (snímačů a akčních členů, atd.) a která je navržena tak aby umožňovala přenos dat v reálném čase, orientovaný na řízení použitím I/O zpráv a přenos informací nižší priority (např. parametry konfigurace zařízení a odlaďovacích dat) použitím explicitních zpráv [24] [23] [25] [34].

Výhodou sítě DeviceNet je také to, že kromě dvou datových vodičů jsou v kabelu (viz Obr. 2.7) obsaženy i dva vodiče přímo pro napájení snímačů a jiných přístrojů. Pro pořádek dodejme, že posledním vláknem (v Obr. 2.7), které je v kabelu uloženo je zpevňovací lanko. [26]



#### Výhody DeviceNetu:

- cenově příznivé řešení (nízké náklady na rozvody a instalace) pro připojení jednotlivých zařízení na síť
- zařízení od různých výrobců, která budou kompatibilní s DeviceNet (tlačítka, startéry, motory, fotobuňky, koncové spínače) lze bez problémů vzájemně kombinovat
- Master/Slave a Peer-to-Peer ke komunikační schopnosti
- DeviceNet je deterministická síť, založená a odzkoušená na známé technologii CAN, což garantuje vysokou spolehlivost sítě



obr. 2.7 DeviceNet: konektor (vlevo), řez vodičem (vpravo)[24]

## 2.8 Uživatelské prostředí SIMATIC Spectation 2.8.2

Vývojové prostředí Spectation je společné pro všechny kamery řady VS720 a je volně k dispozici ke stažení a vyzkoušení. Tento software je založen na hierarchickém uspořádání. Jsou v něm tři základní úrovně: nejnižší je softsensorová úroveň (softsensor level), nad ní je úroveň inspekčních produktů (product level) a nejvyšší je systémová úroveň (system level). Do každé z těchto úrovní patří určité prvky a náleží k ní určité parametry, které lze měnit a další parametry, které lze z ní programovat (zpravidla nižší úrovně).

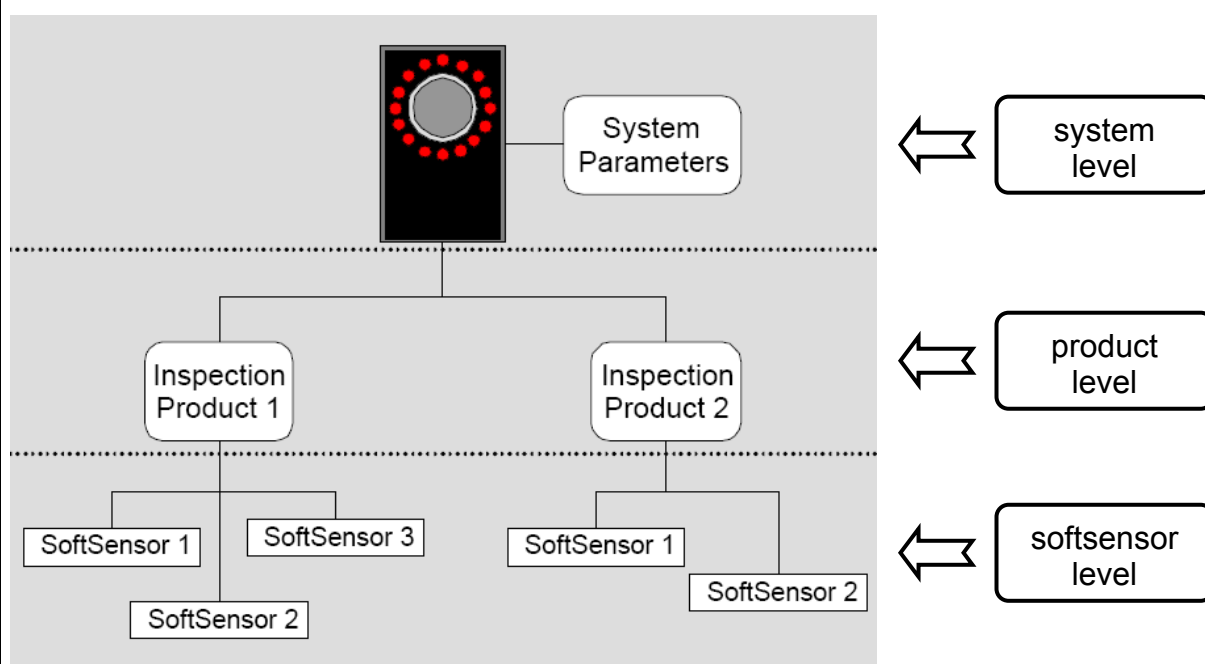
Softsensorová úroveň je nejnižší. Prvky této úrovně jsou softsenzory. Softsenzory jsou základní snímače, které vykonávají různé elementární úkoly jako je zjištění hodnoty intenzity pixelu, detekování hrany, určení plochy, atd. Každý softsensor má své parametry, kterými ho lze nastavit. Hodnoty některých parametrů lze používat nebo měnit z vyšších úrovní prostřednictvím skriptu. Hlavně lze ale softsenzory řetězit tak, že každý vykonává část inspekčního úkolu. Tímto řetězením lze tedy z elementárních softsensorů naprogramovat i složitější inspekční úkol.

Druhou úrovní je úroveň produktová. Naprogramovaný inspekční úkol skládající se z několika softsensorů (případně foreground skriptu) nazýváme inspekčním produktem. Do této úrovně patří i foreground skripty. Z produktové

úrovně (např. ve foreground skriptu) lze používat některé parametry (zpravidla výstupní) úrovně softsenzorové.

Třetí a nejvyšší úroveň je úroveň systémová. Do této úrovně patří i background skript. Ze systémové úrovně lze volat (spouštět) inspekční produkty, případně (přes registry) využívat jimi získané informace. Dále je z této úrovně možné měnit některé parametry pro nastavení celé kamery (doba expozice, režim spouště, nastavení komunikace atd.).

Všechny tři úrovně jsou zobrazeny na obr. 2.8 .



obr. 2.8 struktura programu Spectation [29]

## 2.8.1 Foreground skript

Je to v podstatě softwarový nástroj umožňující psaní krátkých programů. V těchto programech (skriptech) lze využívat informace získané softsenzory.

Potřeba foreground skriptu v podstatě vychází z potřeby dále zpracovávat již získaná data, provádět s nimi aritmetické, logické či jiné operace, číst a ukládat je do registrů pro využití jinými produkty nebo třeba připravovat data k poslání po síti. Skripty jsou proto velice mocným nástrojem při tvorbě netriviálních inspekčních úkolů.

Foreground skript se píše v syntaxi podobné jazyku Java nebo C a tento skript je svázán vždy pouze s jedním konkrétním inspekčním produktem. Je to tedy program běžící na produktové úrovni, který je spuštěn vždy, když je spuštěn daný inspekční produkt.

## 2.8.2 Background skript

Background skript je stejně jako foreground skript program napsaný v syntaxi podobné jazyku Java nebo C.

Background skript na rozdíl od foreground skriptu běží na systémové úrovni (nezávisle na tom zda probíhá nějaká inspekce či nikoli) a není závislý na konkrétním inspekčním produktu. Tento skript běží na pozadí v konečné či nekonečné smyčce. Může být spuštěn třemi způsoby. U prvního způsobu je skript spuštěn při startu kamery. U druhého způsobu je spuštěn signálem a u třetího je spuštěn ručně (využití například při testování).

Pomocí background skriptu je možné provádět předzpracování obrazu, spouštět inspekční produkty, vyčítat data z registrů a nastavovat různé parametry. Především je však pomocí nich možné sledovat stav digitálních vstupů a výstupů a řídit komunikaci kamery s okolím.

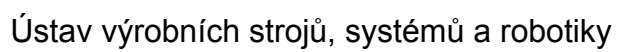
## 2.8.3 Kalibrace kamery

Protože kamera určuje vzdálenosti v pixelech, ale robot potřebuje pro najetí souřadnice v metrických jednotkách (milimetrech) je nutné před tím, než se začne snímat kamerou scénu provést kalibraci kamery. Kalibrace se provádí softwarově na obrazu kalibračního výkresu (viz obr. 2.9) sejmutého kamerou.

Kamera má pro tento účel k dispozici v kategorii nástrojů MathTool softsenzor CoordinateSystem.

Pomocí tohoto softsenzoru vytvořím měřítko, podle kterého se budou dále přepočítávat žádané vzdálenosti. Toto měřítko vytvořím tak, že v softsenzoru CoordinateSystem přiřadím vzdálenostem (souřadnicím) středů terčů v pixelech (na obrazu z kamery) jejich skutečné vzdálenosti (souřadnice) v milimetrech.

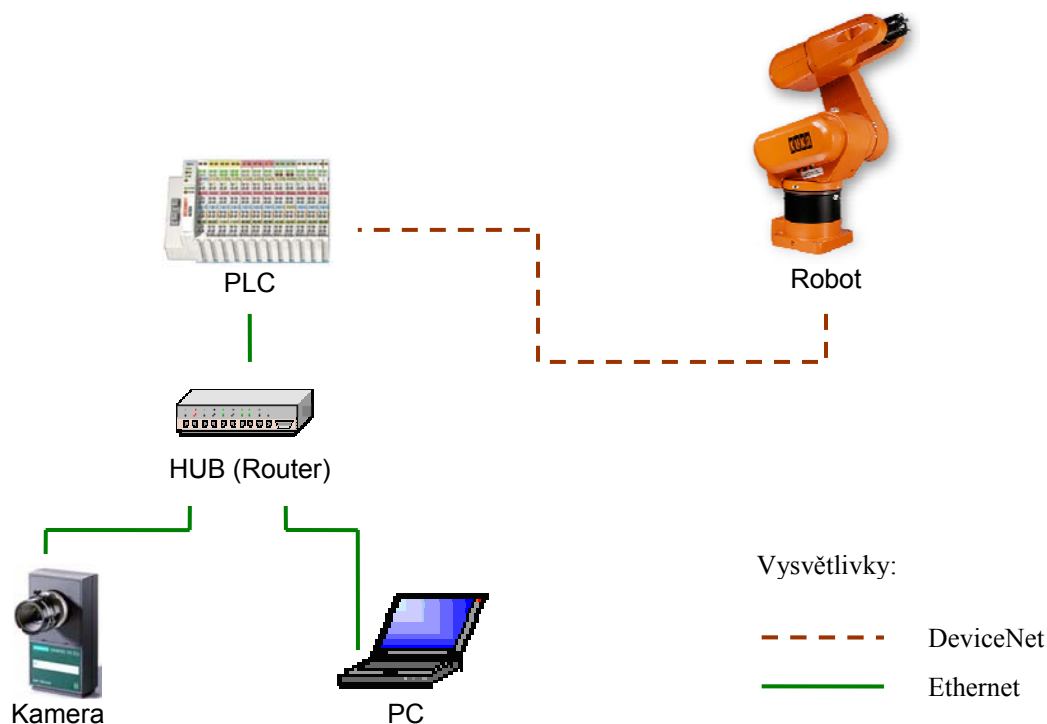




### 3 Koncepce pracoviště

#### 3.1 Schéma

Pro danou demonstrační úlohu bylo zvoleno uspořádání zařízení na pracovišti, znázorněné na obr. 3.1.



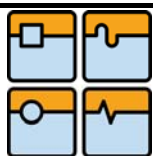
obr. 3.1 Schéma pracoviště pro zadanou úlohu

#### 3.2 Robot

Použitým zařízením je průmyslový robot Kuka KR 3. Jde o robot se 6-ti stupni volnosti, nominální nosností 3 kg (bez koncového efektoru) a přesností polohování  $\pm 0,05$  mm. Pro účely demonstrační úlohy je robot osazen pneumatickou přísavkou, které je k poslednímu členu kinematického řetězce robotu Kuka připevněno přes přírubu. Kamera je připevněna na stejné přírubě.

Robot může v základní konfiguraci komunikovat přes Ethernet (OPC), DeviceNet nebo RS-232.

Při použití sériové linky se však při komunikaci přes ni zruší předpočítávání kroku robotu (standardně robot předpočítává 3 kroky) (funkce advanced), což má za následek nemožnost použít interpolaci. Zrušení předpočítávání kroku je komplikace především u úloh s pohybujícími se objekty, kdy robot musí rychle reagovat a pohybovat se na posílané souřadnice. U ostatních úloh se robot na krátký čas po



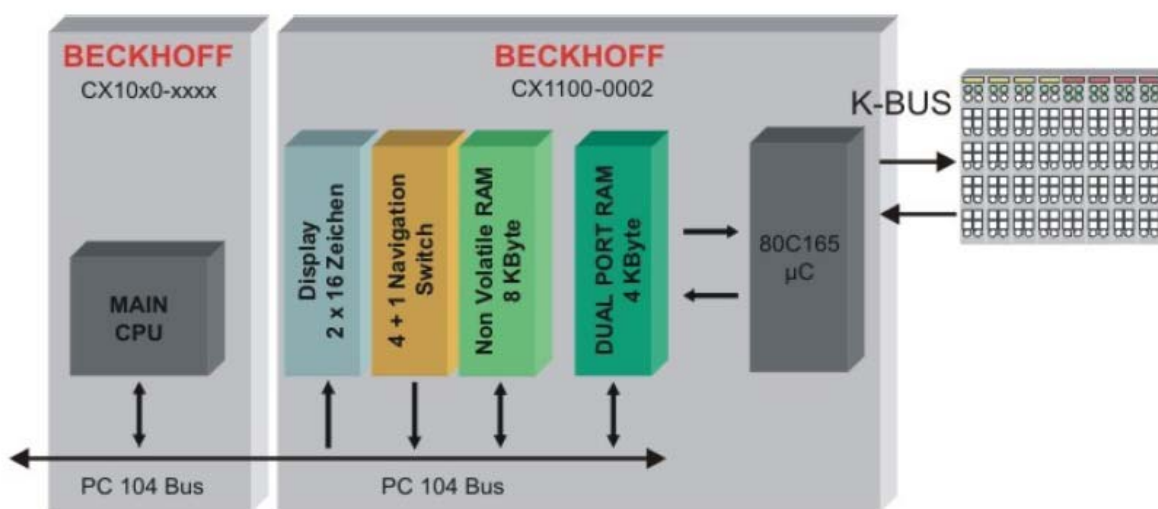
komunikaci pozastaví a funkci advanced příkazem v programu opět zapneme. Během tohoto krátkého času si robot kroky předpočítá a pak může opět využívat interpolaci.

### 3.3 PLC

Použitým zařízením je PLC (Programmable Logic Controller) od firmy Beckhoff. Jde o embedded PC (průmyslový počítač), poskládaný z modulů Beckhoff, které poskytují PLC některé doplňkové funkce.

Základ PLC tvoří modul embedded PC, vybavený operačním systémem Windows CE (operační systém, který je optimalizován pro zařízení, která mají málo místa pro uložení operačního systému. Pro běh jádra systému stačí velice málo místa, v řádech jednotek megabajtů). Modul PC je doplněn o prvky pro indikaci stavu a je připojen k PCI sběrnici (viz Obr. 3.2) (sběrnice PCI se přidala ke standardu PC/104 v roce 1996 [28]). Na tuto sběrnici je pak připojen i modul s ethernetovým a USB výstupy [27].

Pomocí dalších modulů je PLC rozšířeno o paměť (paměťovou kartu), přepínače, digitální vstupy a výstupy a také je PLC umožněna komunikace s okolím přes DeviceNet.



obr. 3.2 Schéma základní architektury PLC [27]

### 3.4 Kamera

Použitým zařízením je inteligentní kamera firmy Siemens, konkrétně kamera SIMATIC VS722A [16]. Tato kamera je vybavena snímacím CCD čipem, který je schopen pořizovat snímky rychlostí až 60 fps. Rozlišení snímku je 640x480 a doba uzávěrky se pohybuje od 10  $\mu$ s do 1 s (elektronická uzávěrka). S ostatními zařízeními kamera komunikuje prostřednictvím Ethernetu a 8 digitálních

vstupů/výstupů. Kromě 64 MB operační paměti má kamera také 16 MB paměti pro programy inspekcí. Tento kamerový systém podporuje komunikační protokoly MODBUS, TCP/IP, Vs Link.



*obr. 3.3 Inteligentní kamera Siemens SIMATIC VS722A [16]*

### 3.5 Popis komunikace

Na základě obr. 3.1 je vidět, že komunikace mezi zúčastněnými zařízeními probíhá po Ethernetu a DeviceNetu.

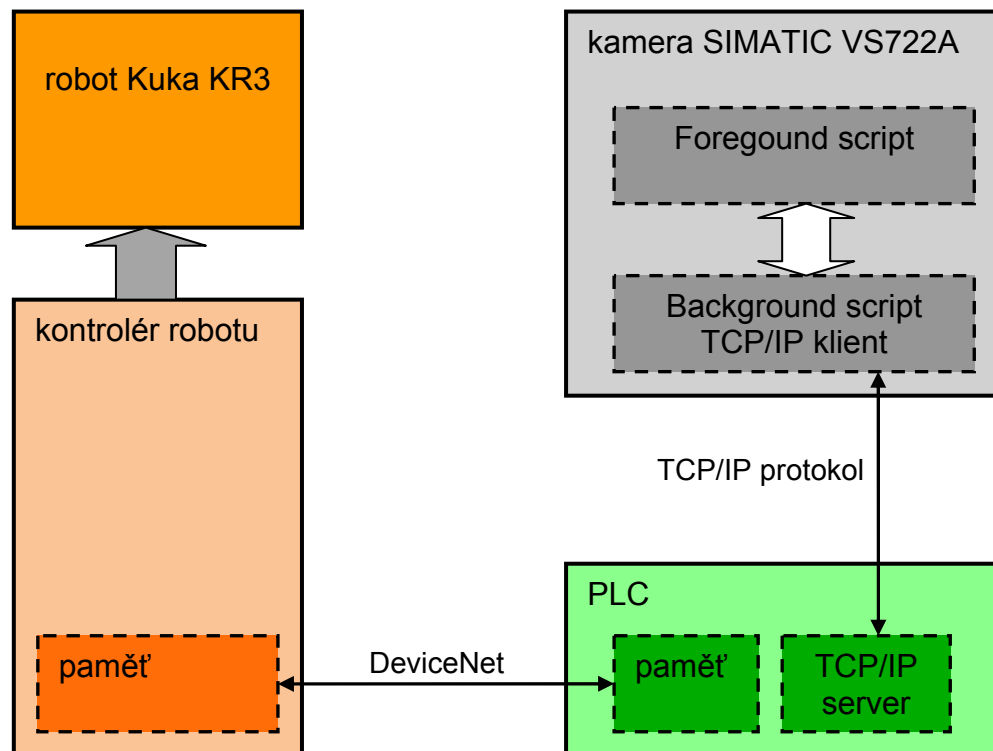
Pomocí komunikačního počítače, připojeného přes router k ostatním zařízením, se nahrají programy do kamery (inspekční program) i do kontroléru robotu (hlavní program) a také do PLC. Při běhu demonstrační úlohy je pak na komunikačním počítači možné sledovat aktuální průběh inspekce.

Komunikace při běhu demonstrační úlohy má dvě hlavní části (viz obr.3.4).

První je komunikace mezi kamerou a PLC. Tato komunikace probíhá přes Ethernet, což vyplývá z možností použité kamery, a je realizována pomocí protokolu TCP/IP. V PLC běží TCP/IP server, který komunikuje s TCP/IP klientem v kameře.

Druhá část je komunikace mezi robotem a PLC. Tato komunikace probíhá přes DeviceNet. Výhodou DeviceNetu je snadná komunikace a výměna dat mezi programem robotu a programem PLC.

U DeviceNetu, v mém případě, může robot pracovat s daty uloženými v PLC, takže odpadne vyčítání dat ze sběrnice (posílání dat do kontroléru robotu). Tímto dosáhnou jednoduššího programování tím, že data (ve formě proměnných) uložená v paměti PLC můžou volat přímo z kódu (programu) běžícího na robotu. K tomuto účelu vyhradím v kontroléru robotu i v PLC kus paměti, kam budou "pracovní" data ukládána.




obr. 3.4 Schéma komunikace

### Alternativní řešení

Jinou variantou komunikace by bylo použití pouze Ethernetu. Tato varianta je podrobněji popsána v [13]. Při této variantě kamera komunikuje s kontrolérem robotu přes počítač (PC). Na PC by běžel TCP/IP server, který by komunikoval s TCP/IP klientem v kameře a předával by data (výsledky inspekce z kamery) OPC klientu (běžícímu na stejném počítači), od kterého by tato data byla předána OPC serveru - kontroléru robotu. Robot by potom na základě těchto dat provedl konečnou manipulaci.

Nejjednodušším řešením komunikace kontroléru robotu s kamerou je přímá komunikace. Tady jsou v zásadě dvě možnosti:

- komunikace přes TCP/IP - tato varianta v zásadě je možná, ale pouze při dokoupení doplňkového softwarového "balíčku" pro robot Kuka, který by umožnil zpracovat TCP/IP pakety v kódu robotu Kuka.

	Ústav výrobních strojů, systémů a robotiky	Str. 30
	DIPLOMOVÁ PRÁCE	

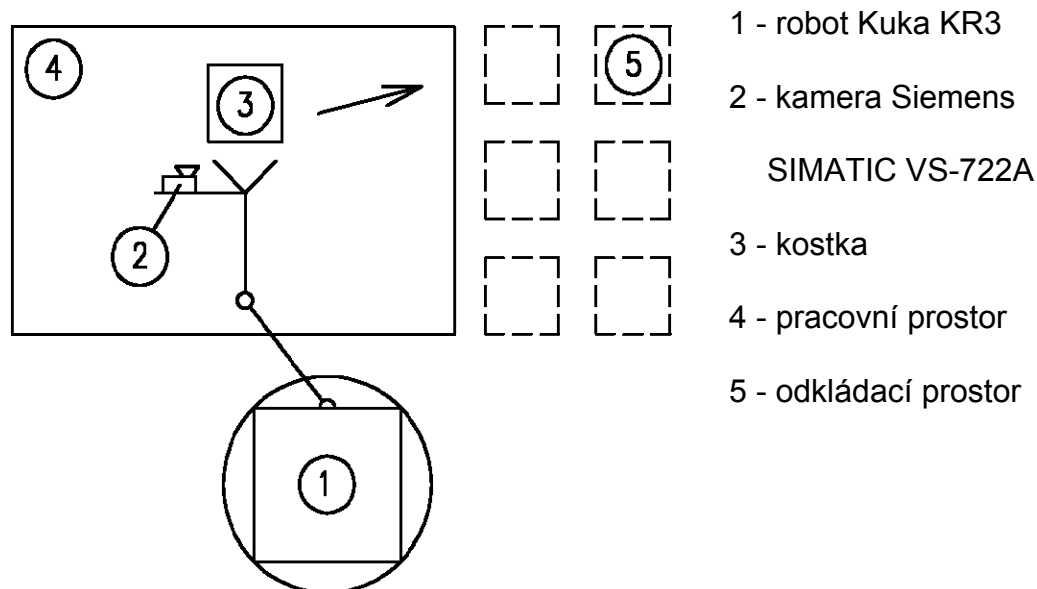
- komunikace přes OPC - tato varianta je v zásadě možná taky. Narážíme zde ovšem na potřebu dokoupení doplňkového "balíčku" pro kameru SIMATIC VS-722A, který by doplnil komunikační možnosti kamery o možnost OPC klienta. Vzhledem k poměrně dlouhé době odezvy (i kolem 200 ms) můžu však říci, že tuto variantu považuji vhodnou pro průmyslové aplikace spíše pro komunikaci na vyšší úrovni (úrovni řídicího systému) než pro úroveň nižší (úroveň zařízení).

## 4 Popis a návrh koncepce řešení

### 4.1 Popis demonstrační úlohy

Cílem demonstrační úlohy je pomocí kamery připevněné ke koncovému efektoru robotu Kuka, sejmut obrázek scény a na základě jeho vyhodnocení provést přesun součástky (kostky) do přesné pozice v odkládacím prostoru. Kostka bude mít v rámci pracovního prostoru libovolnou vstupní polohu a orientaci.

Uspořádání a popis pracoviště, na kterém byla realizována demonstrační úloha je vidět na Obr. 4.1. Toto pracoviště je realizováno v odborné laboratoři na Ústavu Výrobních Strojů, Systémů a Robotiky na Strojní Fakultě.



obr. 4.1 Schéma pracoviště pro zadanou úlohu

### 4.2 Rozbor problematiky manipulace volně ložených prvků

Při řešení problému manipulace volně ložených prvků se v případě mé úlohy můžeme dívat na věc z pohledu:

- objektu
- kamery - a v ní použitých softsenzorů
- robotu Kuka (a koncového efektoru)

Při tvorbě programu pro mou demonstrační aplikaci jsem zvažoval různé aspekty, na základě kterých jsem navrhnul konečné řešení. Tyto aspekty jsou koncepce úlohy, možnosti softsenzorů, uchopení objektu, vzdálenost v ose z mezi kostkou a objektivem kamery a možnosti robotu.

### 4.2.1 Koncepce úlohy podle pohyblivosti objektu

- stojící objekt
- pohybující se objekt

Koncepce programu pro celou úlohu závisí mimo jiné na tom, zda se objekt, který má být manipulován pomocí robotu, pohybuje či nikoli. Tato skutečnost má zásadní vliv při sestavování celkového algoritmu úlohy.

Určení polohy stojícího objektu obnáší pořízení a zpracování snímku. Hlavním cílem je v daných okamžicích zjistit, zda se na scéně vyskytuje hledaný objekt, přičemž může mít kamera naučeno více druhů objektu a při určování jejich polohy provádět i jejich kontrolu a selekci. Pro následnou manipulaci jsou dvě základní možnosti řešení.

V prvním způsobu řešení jsou nalezeny všechny objekty, které se v dané oblasti nalézají, následně dojde k uložení a poslání souřadnic všech objektů a poté následuje manipulace všech nalezených objektů bez najíždění do snímací pozice.

V druhé variantě řešení je přístup takový, že je pořízen snímek scény, poté je vyhodnoceno, zda se v ní nachází alespoň jeden objekt. V případě přítomnosti objektu jsou rovnou poslány jeho souřadnice a následně je provedena jeho manipulace robotem a poté robot najede do stejné snímací pozice. V případě, že se na aktuální scéně nenachází objekt, dojde k přesunu robotu na novou snímací pozici.

Oba přístupy jsou možné a v praxi využívány. Zvolit jeden z nich lze například podle toho, jak rychle přibývají objekty na pracovní plochu. Pro mou úlohu jsem zvolil druhý přístup.

V úlohách, ve kterých se objekt pohybuje je potřeba kromě rozpoznání a lokalizace objektu vypočítat jeho polohu, ve které bude za určitou dobu a zajistit aby do této doby byl na stejném místě i robot. Úlohy tohoto typu proto obnáší pořízení několika snímků, ze kterých je stanovena rychlost objektu, určení jeho polohy a stanovení místa a času uchopení objektu, přičemž je důležitá časová synchronizace robotu a ostatních zařízení.

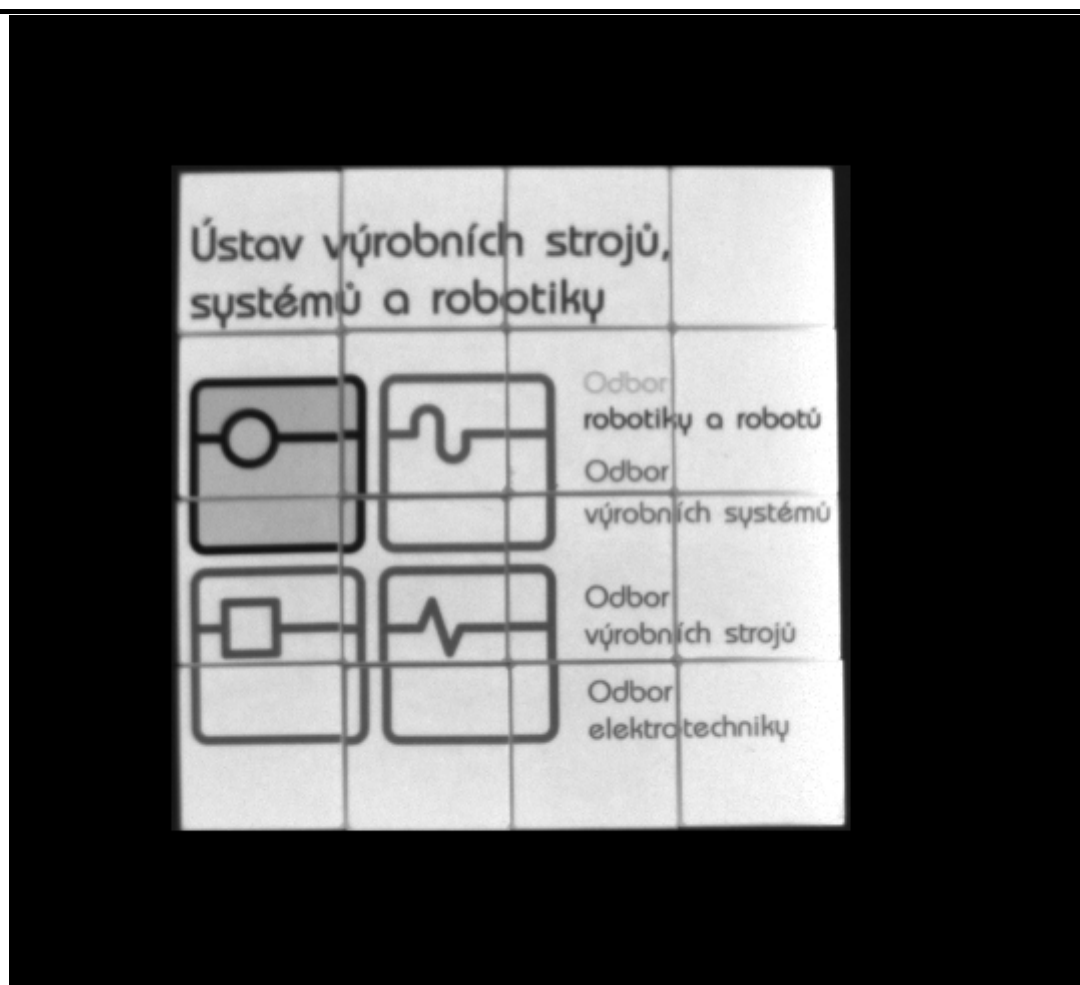
### 4.2.2 Problematika rozpoznávání objektu

Problematiku rozpoznávání objektu v mé úloze jsem rozdělil do několika kroků :

- jeden objekt
- jeden objekt vícenásobně
- obecný geometrický tvar
- objekt s textem

V demonstrační úloze je skládán obrázek loga Ústavu Výrobních Strojů, Systémů a Robotiky (dále jen ÚVSSR)(viz. Obr. 4.2), sestávající se z 16 kostek, z nichž každá obsahuje část tohoto loga.



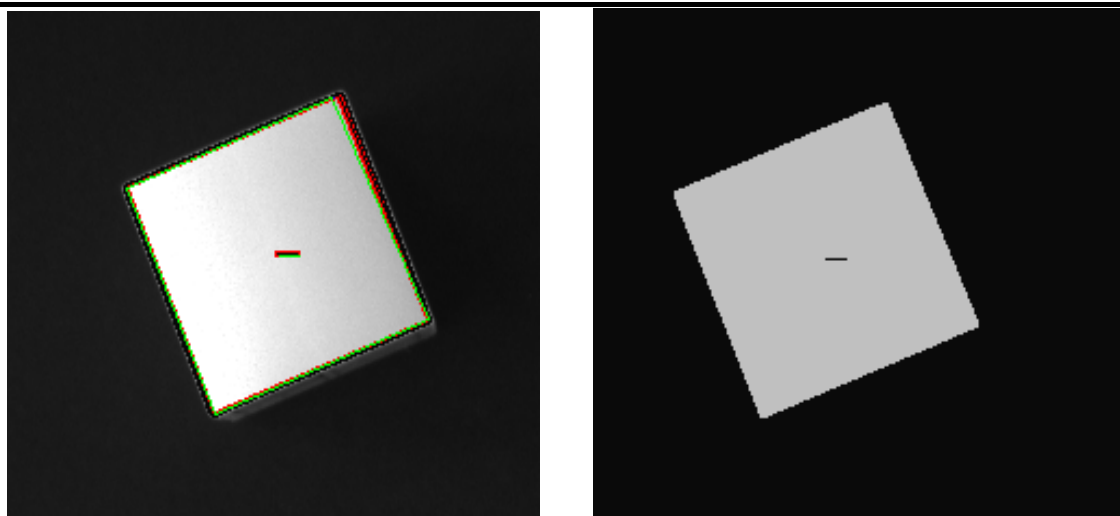


obr. 4.2 Snímek kompletního loga ÚVSSR

Existuje více druhů inspekci. Většinou se snažíme hned od začátku uzpůsobit podmínky tak aby samotná inspekce byla co nejjednodušší a co nejrychlejší. Tato předpříprava potom snižuje požadavky na použitý hardware i programovací složitost programu inspekce.

Při sestavování programu pro moji inspekci se začaly na elementární úloze řešit elementární problémy a postupně se ve více krocích zvyšovala náročnost inspekce.

Pro vyhledávání obecných geometrických objektů (tzn. bez obrázku) lze použít více softsensorů. V rámci testování se nejvíce osvědčily softsensory ObjectFind (viz obr. 4.3 a) ) a BlobTool (softsensorová dvojice BlobGenerator a BlobSelector)(viz obr. 4.3 b) ), které pro základní tvary dávají takřka identické výsledky (co se týče rozpoznání objektu i přesnosti určení jeho polohy). Dále bylo testováno použití softsensoru TemplateMatch.



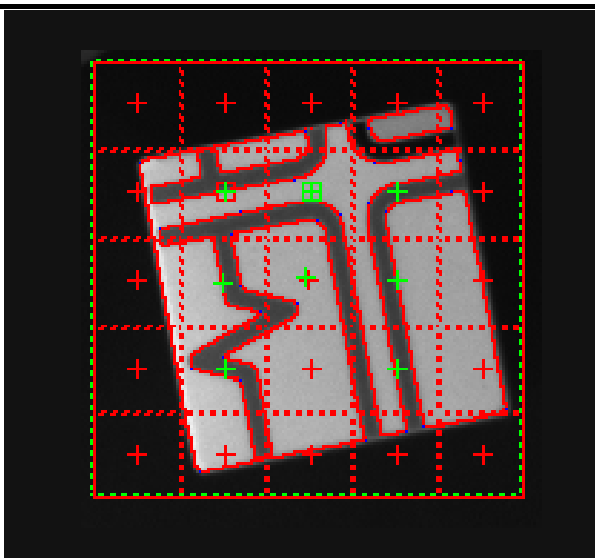
*obr. 4.3 Rozpoznání kostky bez motivu :*

*a) použití softsensoru ObjectFind (vlevo)*

*b) použití softsenzorové dvojice BlobGenerator a BlobSelector (vpravo)*

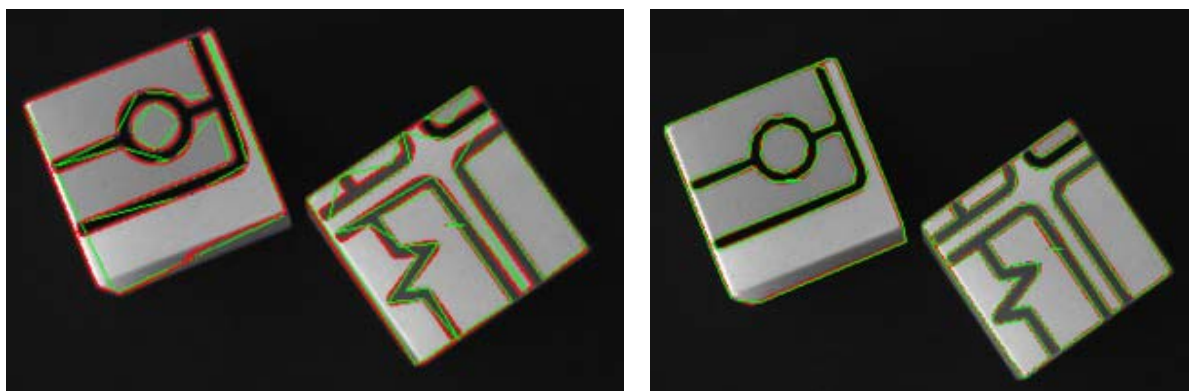
Pro vyhledávání jednoho objektu byl nejdříve použit softsensor ObjectFind, kterému se zadá plocha, ve které má objekt hledat a zadá se mu vzor objektu. Na začátek se hledala kostka s prázdnou horní stěnou - tedy jen čtverec bez obrázku. Pro odstranění nesrovnalostí (zkreslení tvaru) s naučeným vzorem vlivem toho, že objekt neleží vždy přímo pod objektivem kamery, byly použity funkce tolerance měřítka a funkce procentuální shodnost rysů objektu. Takže i když kostka neležela přímo pod objektivem kamery a na snímku byly vidět její boční stěny, tak kostka byla rozpoznána. Softsensor ObjectFind zobrazí na výstupu polohu v ose x a y a také natočení a lze ho použít pro určení polohy (a natočení) všech objektů naučeného vzoru na snímku. Možnost vícenásobných objektů nevyužiji, protože každá kostka má na své horní stěně unikátní část loga ÚVSSR.

Pro vyhledání jednoho objektu bylo také vyzkoušeno použít softsensor TemplateMatch. Tento softsensor se naučí pixelový vzor objektu. Respektive - vzorový objekt rozdělí do polí, z těchto polí vybere oblasti bohaté na detaily (zelené křížky v obr. 4.4) a v těchto oblastech se naučí skupiny (shluky) světlých či tmavých (nebo oboje - je tu možnost volby) pixelů a tyto shluky potom hledá a porovnává s prohledávanou oblastí. Vyhodnocení potom může probíhat na základě míry neshodnosti ploch (procentuální neshodnost ploch) nebo na základě absolutního množství neshodných pixelů. Tento softsensor dokáže určit posunutí, ale jeho zásadní nevýhodou je to, že nedokáže rozeznat objekty, které jsou natočené oproti poloze, ve které byly naučeny (pokud není pomocí skriptu dále zřetězen s některým dalším softsensorem). Z tohoto důvodu jsem se rozhodl použít jiné řešení.



obr. 4.4 Naučení vzoru softsenzorem TemplateMatch

Dalším krokem je rozpoznávání konkrétního motivu. Pro tento úkol je primárně určen softsenzor ObjectFind, kterého bylo použito u většiny kostek (bez textu). Každý motiv vyžaduje (podle konkrétního obrázku) trochu jiné nastavení. V dané úloze se využily především parametry minimální délka části objektu (MinimumLength) a přímota (Straightness). Parametr Straightness ovlivňuje rozpoznání významným způsobem. Při kopírování radiusu se skutečný tvar aproximuje úsečkami. Tento parametr udává délku nejmenší tětiny radiusu a tím ovlivňuje, jak přesně bude učený vzor kopírovat originální tvar. Parametr MinimumLength slouží k podobnému účelu ovšem s tím rozdílem, že jeho hlavní využití je u ne-zakřivených, ale lomených hran. Tento parametr udává nejmenší délku úsečky, která bude rozpoznána jako část objektu (viz Obr. 4.5 a) b) ). Je pravda, že čím složitější bude vzor, tím déle bude trvat jeho rozpoznání, ale naproti tomu se zmenší pravděpodobnost, že "projde" objekt podobného tvaru.

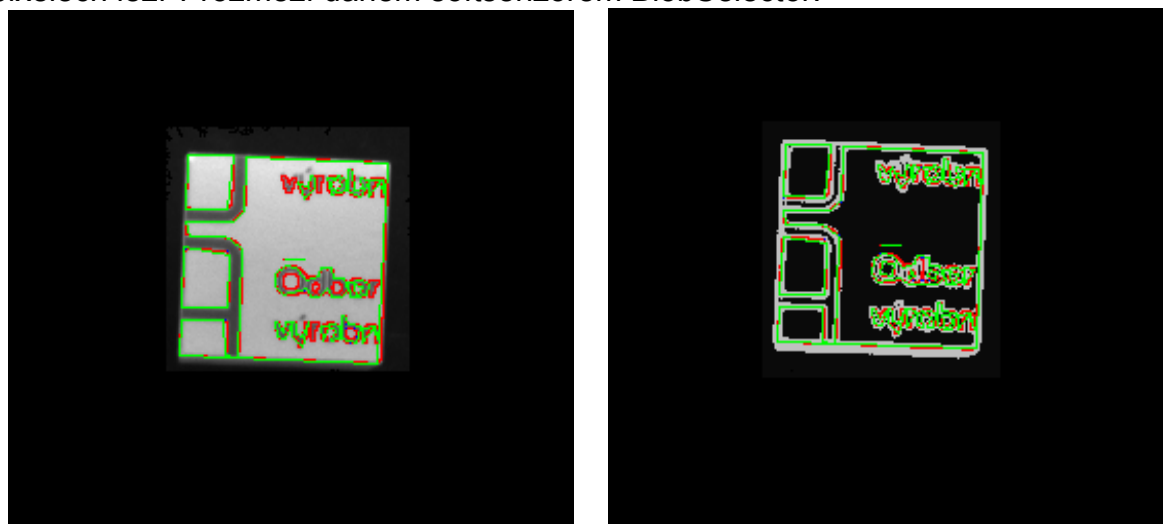


obr. 4.5 Rozpoznání kostky softsenzorem ObjectFind :  
a) standardní nastavení (vlevo)  
b) nastavené parametry MinimumLength a Straightness (vpravo)

Na některých kostkách je kromě geometrického obrázku také text. Proto jsem se zabýval i tím, zda by se právě tento text nemohl stát rozhodovacím srovnávacím

kritériem. Kamera má pro rozpoznávání textu k dispozici nástroj - softsenzor OCR (Optical Character Recognition), který dokáže porovnat text ve zkoumané lokalitě s naučeným vzorem. Jako výstup tohoto softsenzoru tedy je informace, zda se text shoduje se vzorem/vzory (písmen, číslic, atd.) a výstupem je text uložený do proměnné typu string. Tato hodnota může být dále porovnávána ve skriptu. Zásadní nevýhodou je to, že plocha ve které se text hledá je nepohyblivá - tedy prozkoumávaná plocha se neposunuje podle nalezených objektů a musí se tedy tento softsenzor řetězit s dalšími. Další nevýhodou je určení rotace textu. Softsenzor je schopen přečíst i text skloněný pod určitým úhlem, ale není primárně určen k tomu tento úhel počítat. Pokud se tedy kostka vyskytne ve stejné poloze (byť skloněná) tak je text rozpoznán, ale není možné určit natočení objektu podle textu, protože tento úhel není primárním výstupem softsenzoru (tím je porovnání textu s naučeným vzorem). Tento nástroj tedy najde spíš využití v aplikacích, kdy zhruba stejně orientované objekty se pohybují například po pásovém dopravníku (kontrola etiket na stejně natočených lahvích pohybujících se po dopravníku).

Další možností, kterou jsem nakonec zvolil i já, je rozeznávání textu jako obrázku softsenzorem ObjectFind (výstupem tedy není proměnná typu string, ale souřadnice středu objektu a jeho natočení). Ale vzhledem k tomu, že tento softsenzor potřeboval pro správné rozpoznání kvalitnější vzor, tak jsem ho navázal na výsledek softsenzorové dvojice BlobGenerator a BlobSelector. Softsenzor BlobGenerator má totiž v pokročilém nastavení možnost použít pro předzpracování filtr. Použitím filtru se výrazně zvedne kvalita rozpoznávaného motivu - textu. Tato úprava je vidět na Obr. 4.6 b) kdy naučený vzor mnohem lépe kopíruje tvar písmen textu na motivu kostky. Tento vyfiltrovaný motiv je už pak možné použít k naučení vzoru pro softsenzor ObjectFind, který následně opět určí polohu středu kostky (a rotaci) pro následnou manipulaci. Softsenzor BlobSelector byl použit proto, aby do výsledků z výstupu na softsenzoru BlobGenerator byly zahrnuty jen objekty jejichž plocha v pixelech leží v rozmezí daném softsenzorem BlobSelector.



obr. 4.6 Rozpoznání motivu s textem :

a) použití softsenzoru ObjectFind (vlevo)

b) použití softsenzorů BlobGenerator, BlobSelector a následně ObjectFind (vpravo)



### 4.2.3 Problematika uchopení

Z pohledu uchopovacího prvku:

Pro danou úlohu jsem uvažoval o použití pneumatického podtlakového uchopovacího prvku (přísavky) nebo použití pneumaticky ovládaných uchopovacích čelistí. Pro kostky použité v mé úloze by šly použít oba tyto uchopovací prvky.

V případě, kdy se kostka nehýbe lze bez problémů použít pneumaticky ovládané čelisti. Přísavka se s výhodou použije k manipulaci pohybujících se objektů, protože je šetrnější k pohonům robotu v případě drobné odchylky udávané polohy od aktuální polohy. Navíc použitím přísavky odpadnou problémy při paletizaci kostek těsně vedle sebe, protože přísavka drží kostku pouze shora. Nevýhodou přísavky je to, že potřebuje horní poměrně rovnou stěnu o ploše alespoň takové jako je plocha přísavky (což může být problém u složitějších nebo u dutých tenkostěnných objektů).

Vzhledem k povaze demonstrační úlohy jsem se nakonec rozhodl použít pneumaticky ovládané přísavky (viz obr. 4.7). Vzhledem k možnosti použití čelistí jsem však navrhnul postup eventuálního řešení pro software kamery i pro tuto variantu.



obr. 4.7 Použité provedení koncového efektoru pro danou úlohu



Z pohledu kamery:

Při řešení problému manipulace kostek pomocí robotu s pneumaticky ovládanými čelistmi je třeba ošetřit možnost výskytu více kostek na snímané ploše, kdy je třeba zamezit manipulaci v případě, že kolem kostky není dostatek místa pro uchopovací čelisti.

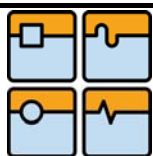
Napadly mne dvě možnosti. V první se spočítají vzdálenosti středu kostky (s číslem jedna, přiřazeným softsensorem ObjectFind)) od všech ostatních středů kostek na ploše. V případě, že některá z těchto vzdáleností bude menší než minimální bezpečná vzdálenost (která je rovna součtu úhlopříčky kostky a bezpečné mezery pro nasunutí čelisti) pro uchopení kostky, tak algoritmus spočítá vzdálenosti pro další kostku (s číslem dva). Až nalezne objekt, který bude dost daleko od okolních objektů, tak předá jeho souřadnice robotu k manipulaci.

Úkol by byl ale ještě zkomplikován tím, že po tomto výpočtu musí následovat rozpoznání přesně tohoto objektu.

V druhé variantě by se nechalo prohledat blízké okolí jedné kostky na přítomnost jakýchkoli jiných objektů a v případě že nebudou nalezeny žádné objekty, tak předám souřadnice kostky robotu. Tento přístup se dá taky realizovat tak (v případě světlé kostky na tmavém pozadí), že hledám v okolí kostky dostatečný počet tmavých pixelů. Současně (dalším parametrem stejného nástroje) musím zajistit minimální vzdálenost středu objektu od hranice prohledávané plochy. Tento přístup je dobře realizovatelný například u kamer firmy Sick, která má pro tento účel vhodnější programovatelné parametry nástrojů. U kamery SIMATIC VS722A by se tyto parametry musely doprogramovat ručně.

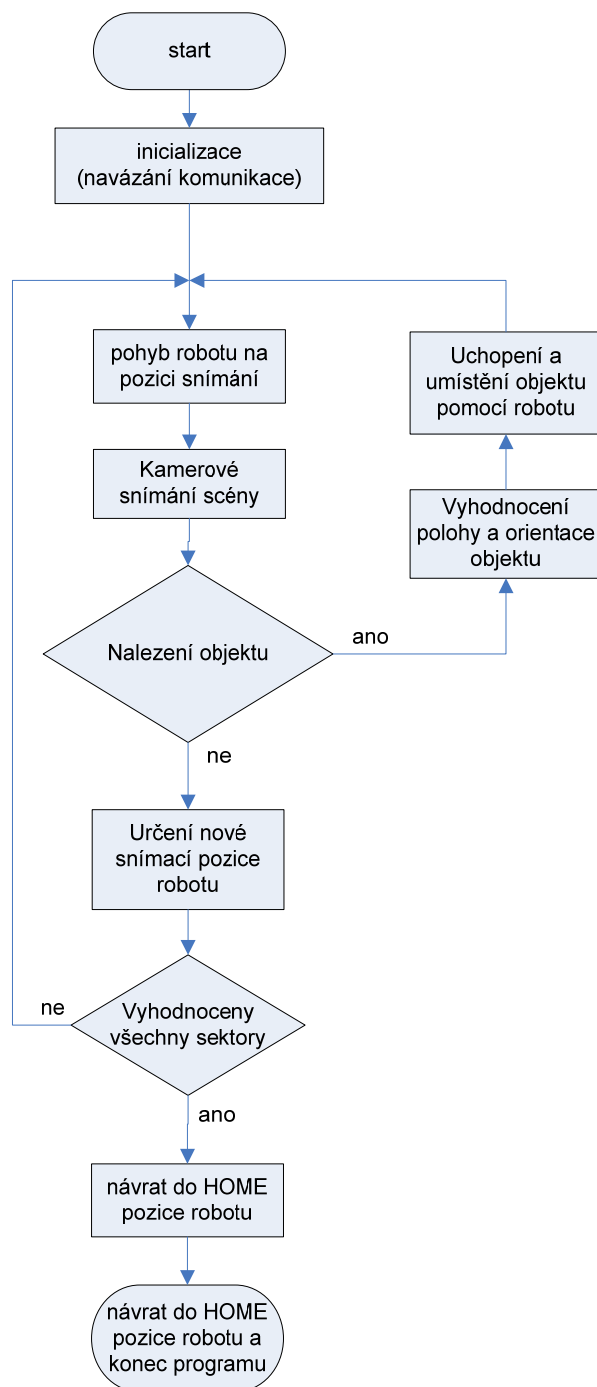
#### 4.2.4 Vzdálenost kamery od pracovní plochy

V závislosti na svislé vzdálenosti (osa z robotu) objektivu kamery od pracovní plochy lze měnit velikost vyhodnocené plochy připadající na jeden snímek. Čím větší bude tato vzdálenost, tím menší budou objekty a budou se hůře rozpoznávat detaily objektu (klesne spolehlivost rozpoznání - jeho opakovatelnost). Naopak - při menší snímací vzdálenosti se budou snadněji rozpoznávat detaily, ale na prozkoumání pracovní plochy bude potřeba několik snímacích pozic. Vzhledem k potřebě rozpoznání kostek obsahující text o malé výšce (názvy odborů - viz obr.4.2). Vzhledem k tomu, že během činnosti programu nelze měnit vzdálenost v ose z (objektiv není vybaven funkcí automatického zaostřování) bylo rozhodnuto, že se bude scéna snímat z menší výšky a celá pracovní plocha bude postupně projížděna a budou se na ní hledat objekty do té doby dokud buď robot nenajde všechny kostky nebo neprojede všechny sektory nebo mu bude vydán pokyn k ukončení úlohy. Sektory se musí překrývat o vzdálenost úhlopříčky kostky, protože objekty které neleží v pořízeném snímku celé nejsou rozpoznávány. V programu robotu se k účelu posunutí snímaného pole použije funkce geometrické operátor v jazyce KRL. Pomocí geometrických operátorů můžu naprogramovat relativní vztahení aktuální snímací báze k jiné uživatelem definované bázi (mimo jiné i relativní vztahení pozice ke zvolené bázi).



## 5 Demonstrační úloha

Schéma celé úlohy a její postup je uveden na obr. 5.1. Zadanou demonstrační úlohu můžeme rozdělit na několik částí, přičemž ke každé této části náleží vlastní program. Patří sem program inspekce v kameře, program klienta (v kameře), program serveru v PLC, program činnosti PLC a program robotu.



obr. 5.1 Vývojový diagram demonstrační úlohy

Úloha bude probíhat takto: Nejdříve se zapnou všechny periferie a proběhne inicializace jednotlivých zařízení. Po startu úlohy nejdříve kamera vytvoří komunikační socket a připojí se na danou IP adresu. Tato komunikační cesta bude otevřena v průběhu celé úlohy pro posílání paketů z kamery do PLC (uzavře se po vyhodnocení posledního sektoru). Zařízením, v jehož taktu bude probíhat celá úloha je robot. Robot tedy nejdříve najede do polohy pro snímání obrazu. Při jejím dosažení zapne signál pro kameru. Kamera sejme aktuální obraz a provede na něm inspekci. Pokud dojde k nalezení a rozpoznání objektu ve snímaném sektoru, tak kamera uloží jeho souřadnice a pošle je do PLC. Odtud si je načte robot a na jejich základě provede manipulaci objektu do odkládacího prostoru. Po ukončení manipulace vydá signál pro PLC o ukončení manipulace. Pokud nedojde k nalezení objektu (scéna je prázdná), tak se robot přesune do snímací pozice pro následující sektor. Tímto způsobem postupně prozkoumám celý pracovní prostor. Po prozkoumání posledního sektoru najede robot do své HOME pozice a úloha je ukončena.



*obr. 5.2 Ilustrační foto demonstrační úlohy*



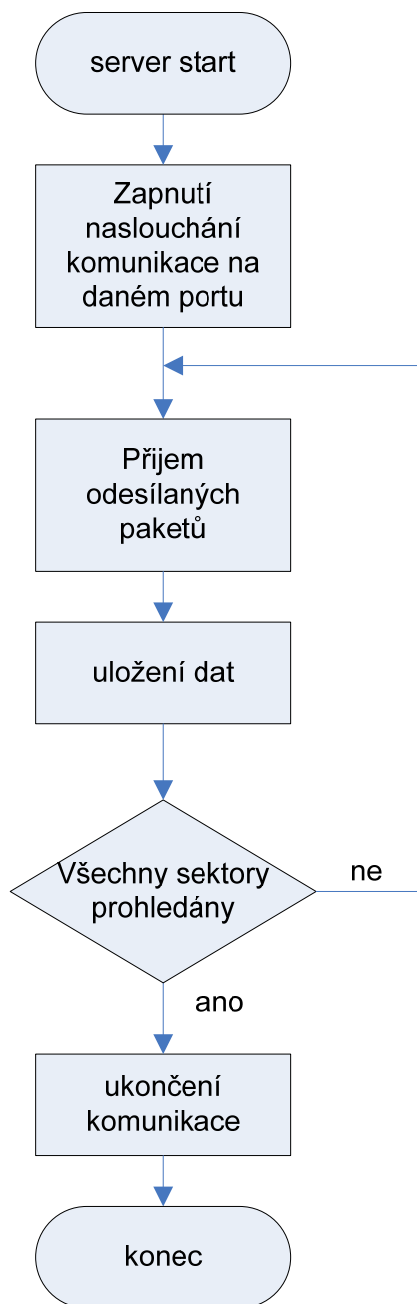
## 5.1 Inspekce

Celá inspekce jednotlivých snímků pořízených kamerou je popsána v inspekčním produktu. Jeho hlavní části jsou foreground skript a softsenzory. Postup inspekce (daný foreground skriptem) je následující.

V daný časový okamžik (dosažení snímací pozice) vydá robot signál PLC, PLC na tento signál zareaguje tím, že vyšle signál k pořízení snímku a následnému spuštění inspekčního produktu (čímž dojde ke spuštění foreground skriptu). Inspekce proběhne tak, že softsenzory vykonají naprogramovanou činnost a její výsledky dají k dispozici skriptu. Foreground skript se zeptá na výsledek softsenzoru ObjectFind. V případě hodnoty PASS (softsenzor našel objekt), provede skript uložení souřadnice  $x$ , souřadnice  $y$  a natočení středu nalezené kostky a přiřazení unikátního čísla kostky. Natočení (úhel pootočení) se vztahuje vždy k poloze, ve které byl objekt naučen. Souřadnice  $x$  a  $y$  jsou vztaženy k levému hornímu rohu softsenzoru. Následuje transformace souřadnic (z hodnot v pixelech na hodnoty v milimetrech), jejich zaokrouhlení a převedení datového typu z typu double (výstupní datový typ ze softsenzoru) na string. Potom tyto řetězce (stringy) vloží do jediného řetězce s názvem paket, přičemž jednotlivé hodnoty jsou oddělené středníkem. Celý řetězec paket pak uloží do registru. Do jiného registru uloží bit s hodnotou 1, který bude použit k určení, zda byl daný paket už jednou odeslán. V případě nenalezení kostky (výsledek softsenzoru FAIL) uloží do proměnné paket string 'objektNenalezen', uloží do registru a opět zapíše hodnotu rozpoznávacího odesílacího bitu. Z registrů si již data dle potřeby načte Background skript.

## 5.2 Popis komunikace server a klient

Jak bylo zmíněno výše, komunikace mezi kamerou a PLC se realizuje po Ethernetu způsobem klient-server. Klientem je background skript v kameře, serverem program v PLC.

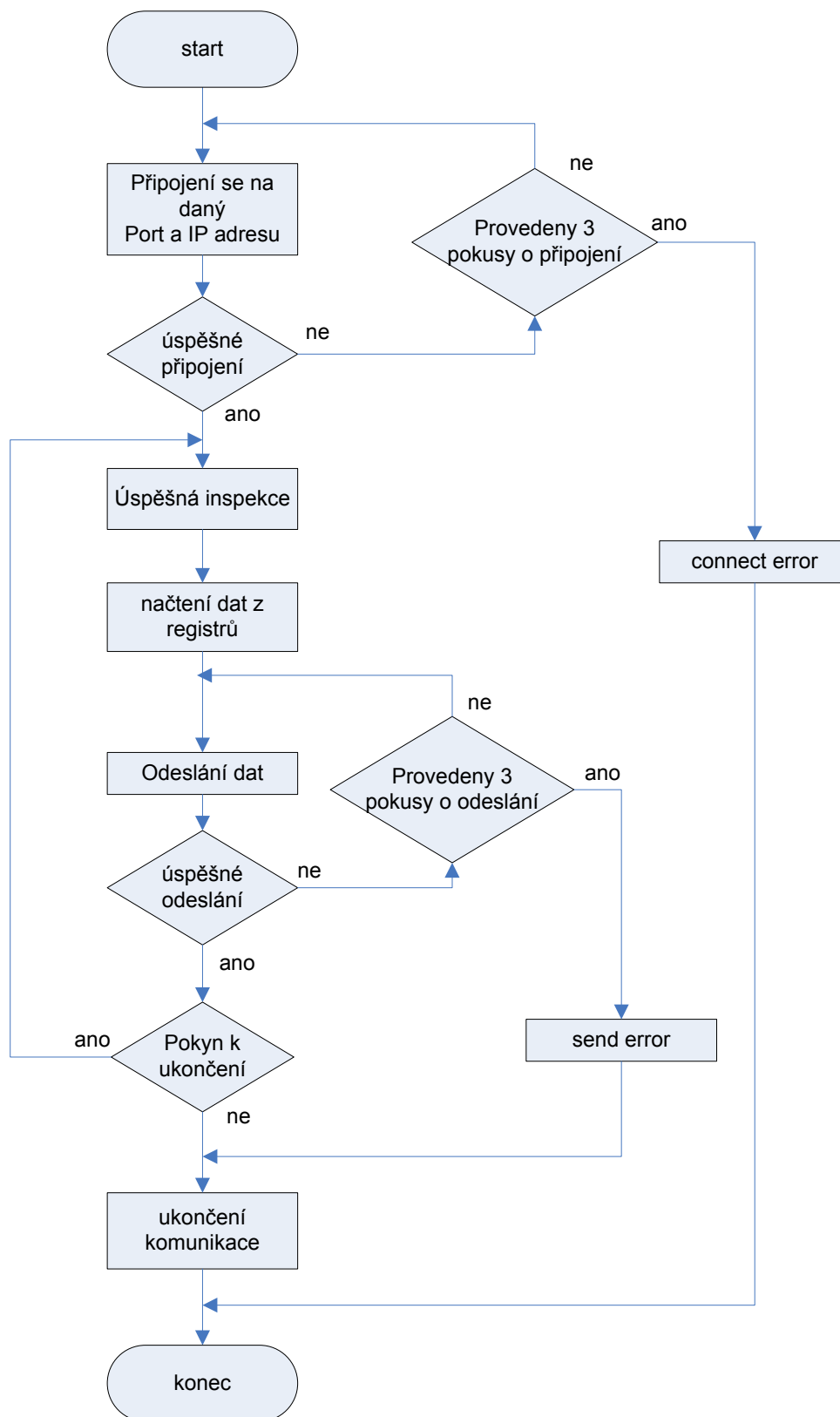
*obr. 5.3 Vývojový diagramu činnosti serveru*

Tato komunikace v dané demonstrační úloze probíhá v několika krocích. Na jeden server může být napojeno více klientů. Zde však bude předpokládáno, že klient je pouze jeden. Obecný postup serveru je následující. Server začne naslouchat komunikaci na určitém portu a čeká až mu přijde požadavek na spojení od klienta. Jakmile mu tento požadavek přijde, uzavře všechny staré spojení a vytvoří na daném portu nové komunikační spojení (socket). Jakmile je socket vytvořen, může dojít k

poslání dat ve formě paketu. Server tento paket přijme a uloží ho. Zpravidla ještě následuje zpětná zpráva klientu o přijetí daného paketu. Tím komunikace končí a server socket uzavře.

V dané demonstrační úloze, jak bylo řečeno výše, je serverem program v PLC (a pracuje podle diagramu na obr.5.3). Po zapnutí serveru začne server naslouchat komunikaci na portu 200. Po vytvoření spojení (socketu) na základě požadavku od klienta v kameře přijímá pakety, které jsou posílány jako pole bytů a ukládá je do paměti v datovém typu string. U serverů s více připojenými klienty je potřeba komunikaci ukončovat a znovu navazovat po každém odeslaném (přijatém) paketu. Protože v mé úloze je jen jeden klient tak jsem se v konečném řešení rozhodl uzavřít socket až na konci úlohy. Celá úloha tedy probíhá v rámci jediného spojení.

Klient je background skript. Při spuštění kamery se spustí background skript (viz obr. 5.4). Nejdříve dojde k vytvoření komunikačního socketu. Poté se skript pokusí připojit k dané IP adrese na daném portu. Aby nedošlo k zacyklování kamery, tak k pokusu o vytvoření a připojení (navázání komunikace) dojde maximálně třikrát. Po navázání komunikace je kamera připravena pro vykonávání úlohy. Skript se začne cyklicky ptát na hodnotu kontrolního bitu (viz podkapitola 5.1). Tento bit má hodnotu 1 při zapsání nových dat a po jejich odeslání se hodnota bitu (registru) přepíše na 0. V podstatě tedy background skript čeká na signál pro získání a vyhodnocení snímku. Pokud hodnota kontrolního bitu je 1, skript načte z registru řetězec (datového typu string), který je určen k odeslání a jako pole bytů ho odešle. Následně je přepsána hodnota kontrolního bitu na 0. Při splnění podmínky pro ukončení úlohy pak program vyběhne ze smyčky a zavře komunikační socket.



obr. 5.4 Vývojový diagramu činnosti klienta



### 5.3 Popis činnosti PLC

PLC má v demonstrační úloze důležitou roli. Přijímá a předává pakety, přijímá a vysílá signály důležité pro koordinaci činností celé sestavy zařízení. PLC je řízeno programem, který se skládá ze dvou základních částí. Jednou je program serveru, druhou je program, který řídí chod PLC. Oba tyto programy běží v PLC současně vedle sebe. Program serveru je popsán v podkapitole 5.2.

Program řídící chod PLC probíhá dle vývojového diagramu na obr. 5.5. Po zapnutí PLC provede program inicializaci proměnných a čeká na zapnutí tlačítka start. Po sepnutí tlačítka start je dán digitálním signálem povel robotu k přesunu do snímací pozice. Při dosažení snímací pozice čeká PLC na signál od robotu snímací pozice dosažena. Při přijetí tohoto signálu od robotu dojde k vyslání digitálního signálu o šířce 10ms do kamery, což spustí pořízení a vyhodnocení aktuálního snímku. Po vyslání spouštěcího signálu začne PLC sledovat stav zaneprázdněnosti kamery. Jakmile stav zaneprázdněnosti (kamera snímá nebo vyhodnocuje obrázek) pomine (sestupná hrana na signálu od kamery do PLC), PLC počká krátkou pevně stanovenou dobu než dojde k poslání paketu. Po této krátké době se zeptá programu serveru na aktuální hodnotu proměnné, do které se ukládají přijatá data a její hodnotu si zkopíruje do své proměnné, kterou použije pro rozklad paketu.

V programu PLC používám 3 pakety. První reprezentuje hodnotu paketu přijatého při předchozí inspekci. Druhý obsahuje hodnotu paketu přijatého po právě proběhlé inspekci a je určen k rozkladu paketu (pracovní paket). Třetí paket také obsahuje hodnotu paketu po právě proběhlé inspekci, ale je jeho zálohou pro následující iteraci.

Následuje rozklad pracovního paketu. Každý přijatý paket je datového typu string a obsahuje 4 čísla oddělené středníkem (za posledním číslem je také středník). Rozklad se provádí v cyklu for (čtyřikrát) a to tak, že se nejdříve najde pozice prvního středníku zleva. Potom se z paketu zkopíruje počet znaků o jedna menší než je pozice středníku, do jednoho prvku pomocné proměnné typu pole stringů (typ pole byl zvolen proto, aby se s částmi původního paketu lépe pracovalo v cyklech). Následně se tento prvek převede na datový typ integer a z pracovního paketu se smaže zleva počet znaků odpovídající pozici prvního středníku. Tento postup se provede čtyřikrát, přičemž po poslední iteraci zůstane hodnota pracovního paketu prázdná. Výsledkem je pole čtyř integerů, které odpovídá souřadnici x, souřadnici y, natočení a unikátnímu číslu kostky.

Po rozkladu paketu následuje rozhodnutí, zda byla nalezena kostka. Navrhnul jsem a vyzkoušel dva postupy. V prvním se při výsledku inspekce FAIL posílá paket o obsahu 'objektNenalezen'. Při rozhodování se pak program ptá, jestli má přijmutý paket tuto hodnotu. Tento postup připouští poslání paketu z kamery při výsledku inspekce PASS i FAIL. Druhý vyzkoušený postup předpokládá poslání paketu pouze v případě výsledku inspekce PASS. Pracuje tak, že se určí počet znaků přijatého paketu. Následně se porovná tento určitý počet znaků v paketu aktuálním a v paketu z minulé inspekce a v případě shody je vydáno rozhodnutí kostka nalezena.

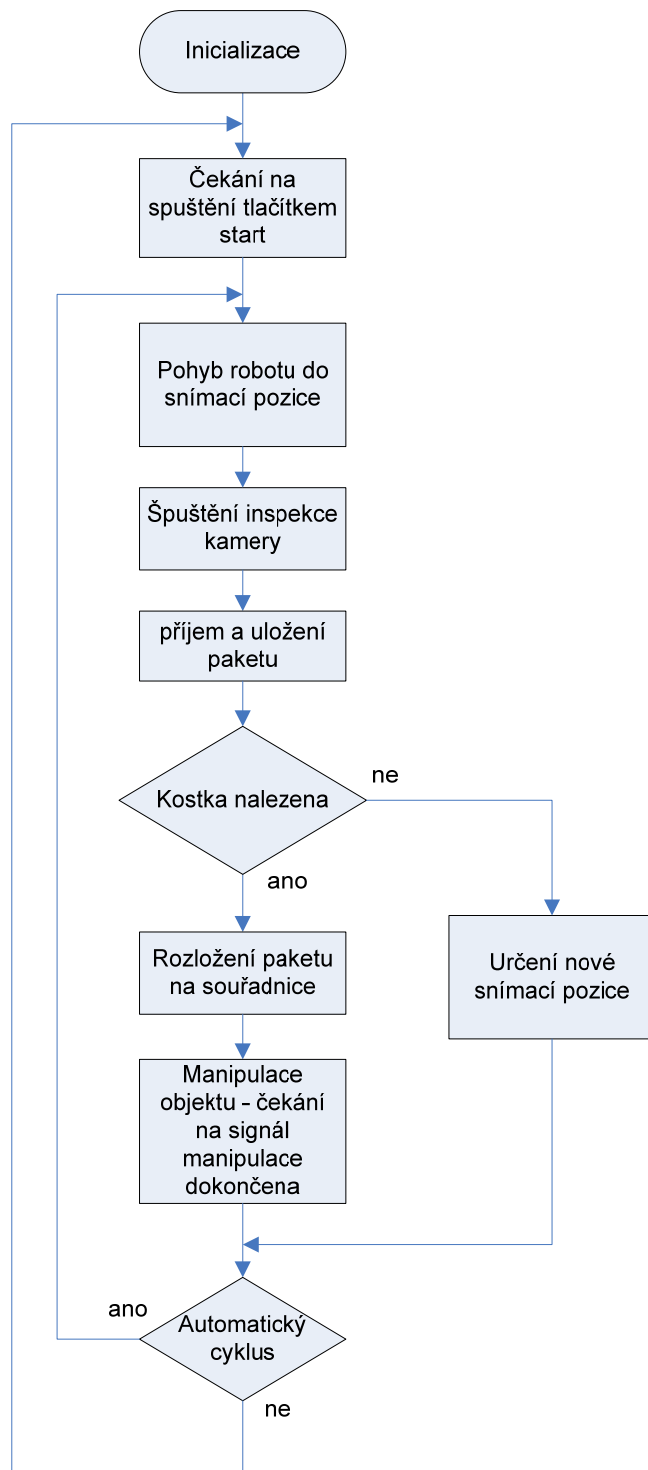
V případě, že nebyla nalezena kostka, vyšle PLC povel robotu k určení nové snímací pozice a čeká, až mu od robotu přijde signál, že tato operace byla provedena.



## DIPLOMOVÁ PRÁCE

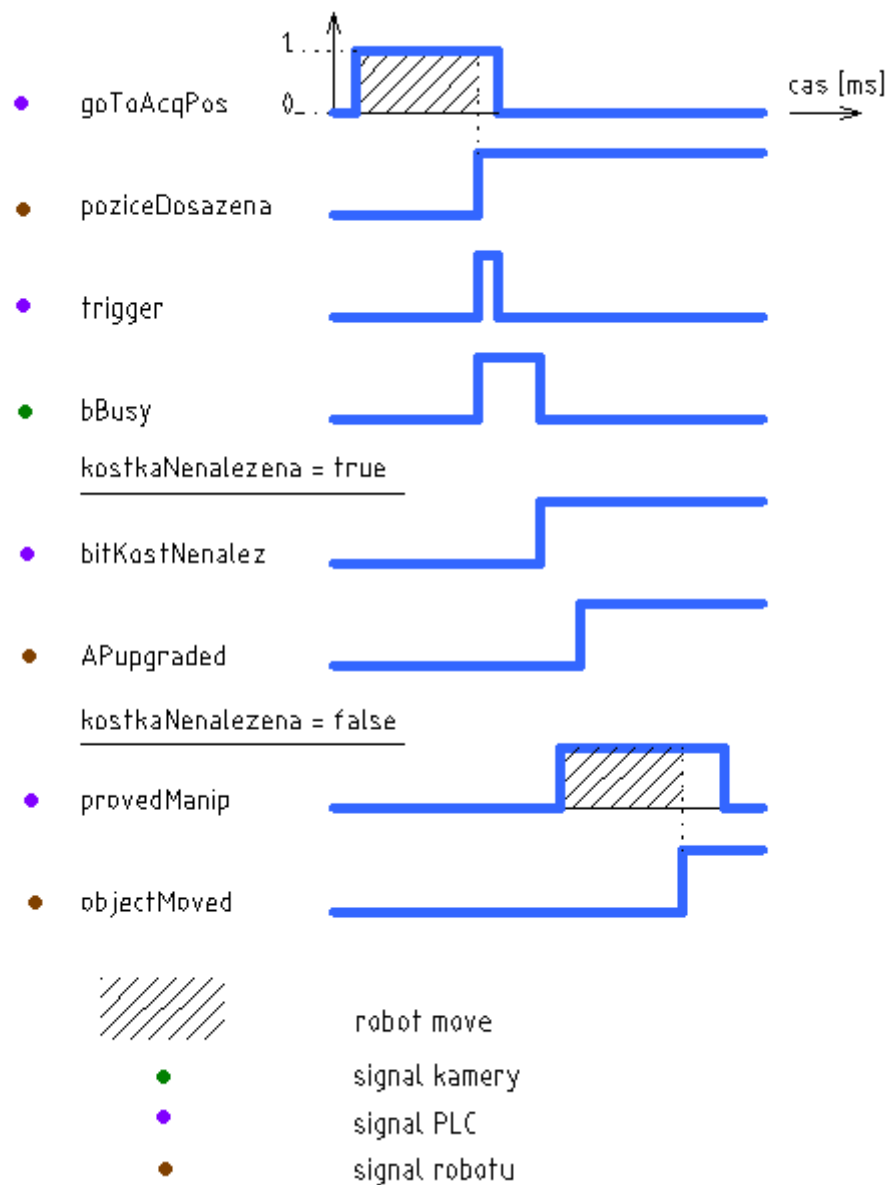
V případě nalezení kostky předá její souřadnice robotu a vyšle mu pokyn k provedení manipulace. Pak čeká na signál od robotu manipulace ukončena.

Pokud je páčkový přepínač na PLC nastaven do polohy automatický režim, následuje přesun robotu do snímací pozice (stejně nebo nové). V opačném případě program ukončí aktuální cyklus, ukončí program a skočí na začátek a čeká na opětovné stlačení tlačítka start.



obr. 5.5 Vývojový diagram činnosti PLC

Činnost PLC je provázána digitálními signály. Jejich průběh je znázorněn na obr. 5.6. V programu pro PLC jsou tyto signály využity jednak k vydávání pokynů pro spouštění kamery nebo pohyb robotu a jednak jako vstupní instance pro zahájení vykonávání některých kroků programu (viz Příloha 3).



obr. 5.6 Průběh digitálních signálů mezi kamerou, PLC a robotem

## 5.4 Popis činnosti robotu

Robot je výkonným členem celé sestavy. Všechny programy a ostatní zařízení v podstatě vykonávají své úkoly, aby mohly robotu předat žádané informace.

Pro danou demonstrační aplikaci byla při tvorbě programu pracovní plocha rozdělena do 8 sektorů, přičemž počet těchto sektorů lze jednoduše změnit (ne za běhu úlohy). Základní funkce robotu je tedy ta, aby postupně (po sektorech) projížděl pracovní plochu. Nad každým sektorem aby se zastavil (kvůli pořízení snímku scény), v případě nalezení kostky provedl manipulaci a najel na stejnou snímací pozici. V případě nenalezení kostky, aby se pak robot přesunul do snímací pozice pro další sektor. Činnost robotu je (jak bylo zmíněno v kapitole 5.3) provázána s činností PLC prostřednictvím digitálních signálů.

Program robotu (viz obr. 5.7) začíná inicializací a nájezdem do HOME pozice. Odtud se na pokyn PLC přesune do snímací pozice (snímacích pozic je stejně jako snímaných sektorů 8). Pro určení nájezdových pozic byla použita jedna základní báze, definovaná uživatelem, a všechny body (pozice) jsou pak vztaženy k ní pomocí funkce geometrický operátor.

Při dosažení snímací pozice spustí signálem (přes PLC) kameru a čeká na informaci (signál) od PLC, zda byla nalezena kostka. Tento signál je vydán v okamžiku zpracování dat z kamery. Robot tedy čeká na signál z PLC o zpracování dat z kamery. Ve chvíli, kdy mu dojde tento signál, se zeptá, zda byla nalezena kostka či nikoli.

V případě, že je kostka nalezena tak robot (DeviceNet master – kontrolér robotu) převezme souřadnice objektu (od PLC – DeviceNet slave), a provede manipulaci. Po manipulaci vyšle robot signál o ukončení manipulace.

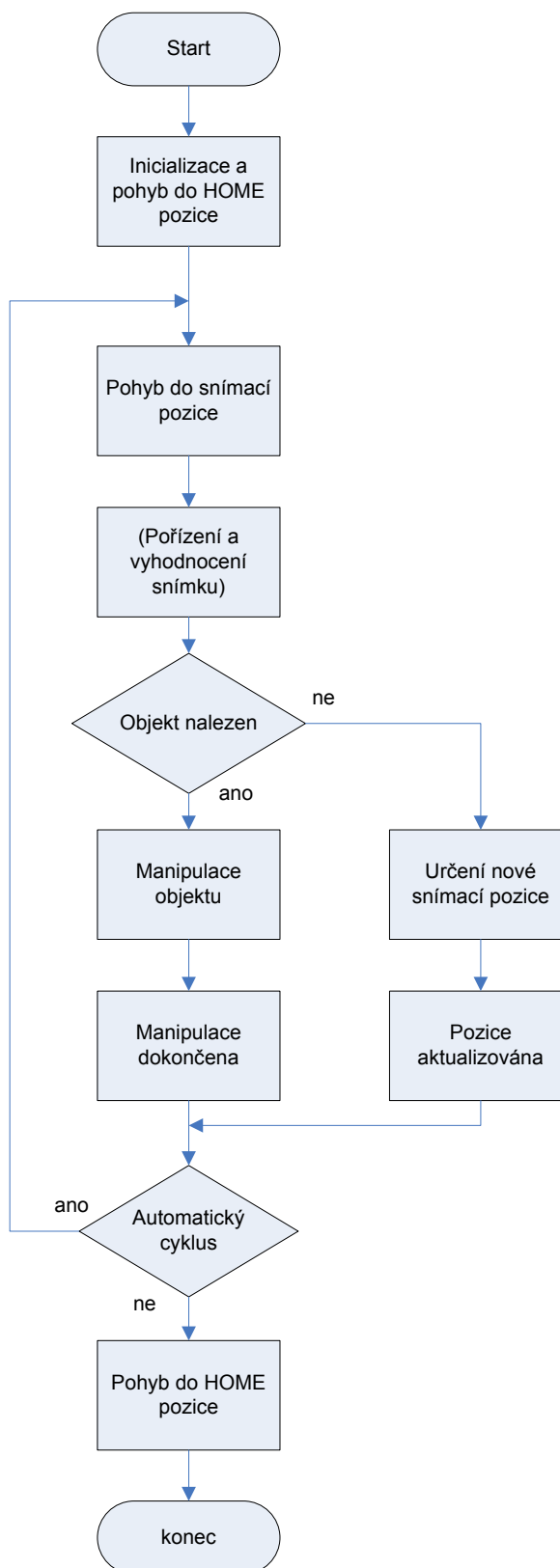
V případě nenalezení kostky, robot zaktualizuje souřadnice snímací pozice a vydá PLC pokyn, že tato aktualizace je dokončena.

Při práci v automatickém cyklu následuje pohyb do snímací pozice a další kolo programu. Pokud je automatický cyklus vypnut, tak se robot přesune do HOME pozice a program končí.





# DIPLOMOVÁ PRÁCE



obr. 5.7 diagram činnosti robotu

## 6 Závěr

Hlavním cílem této diplomové práce bylo seznámit se s principy robotického vidění a vytvořit praktickou aplikaci, která by demonstrovala využití průmyslové kamery Siemens v součinnosti s průmyslovým robotem KUKA.

Nedílnou součástí návrhu celé aplikace byla i tvorba programů pro jednotlivá zařízení. Ve všech případech bylo ve velké míře využíváno simulačních prostředí, což velkou měrou urychlilo tvorbu inspekčního produktu.

Jako řídicí prvek celé aplikace bylo použito PLC Beckhoff, které mimo jiné také zajišťuje výměnu dat mezi kamerou Siemens a robotem KUKA.

Při tvorbě programu pro průmyslovou kameru se jako významný faktor projevila volba softwarových inspekčních modulů (tzv. softsenzorů). U nejkomplicovanějších objektů (kostka s šedým textem o malé velikosti) se nejvíce osvědčilo použití softsenzorové dvojice BlobGenerator a BlobSelector a následné využití softsenzoru ObjectFind. Tato varianta přinesla nejmenší zkreslení výsledku (největší pravděpodobnost správného rozpoznání).

Co se týče komunikace mezi průmyslovou kamerou a PLC, tak se jako ekvivalentní jevil přístup kdy, program serveru TCP/IP byl zakomponován do hlavního programu PLC a přístup kdy program serveru běžel samostatně jako vedlejší program. Dále bylo nutné v PLC také vyhodnotit, zdali souřadnice nalezeného objektu jsou správné. Pro tento účel byly vyzkoušeny dva přístupy. První je schopen vyhodnotit správnost údajů i v případě, kdy není znám aktuální výsledek obrazové inspekce. V druhém případě získá PLC tuto informaci podle výsledku obrazové inspekce (PASS nebo FAIL). Oba přístupy se projevily jako ekvivalentní z hlediska implementace, avšak druhý přístup je méně náročný z pohledu programování.

V rámci vypracování teoretické části jsem zlepšil své vědomosti o systémech vidění. Při sestavování programů pro použitá zařízení jsem si prohloubil své znalosti o síťové komunikaci a zlepšil znalosti v programování robotu, programování kamery i programování PLC.

Věřím, že vědomosti a zkušenosti získané při tvorbě méj diplomové práce využiji i ve svém dalším profesním životě.

## Použitá Literatura


- [1] *Data translation : How To Plan Your PC-Based Machine Vision System* [PDF dokument]. 2003 [cit. 2010-04-27]. Data translation. Dostupné z WWW: <[www.datatranslation.com/resources/How-To-Plan-MV-System.pdf](http://www.datatranslation.com/resources/How-To-Plan-MV-System.pdf)>.
- [2] National Instruments [online]. 2004, 2010 [cit. 2010-04-27]. *Machine vision systems*. Dostupné z WWW: <<http://sine.ni.com/nips/cds/view/p/lang/en/nid/10008>>.
- [3] OMRON [online]. 2007, 2010 [cit. 2010-04-27]. Vision sensors. Dostupné z WWW: <<http://www.ia.omron.com/product/29.html>>.
- [4] OMRON [online]. 2007, 2010 [cit. 2010-02-27]. Zfv. Dostupné z WWW: <[http://www.ia.omron.com/product/family/1523/index\\_fea.html](http://www.ia.omron.com/product/family/1523/index_fea.html)>.
- [5] SIEMENS [online]. 2008, 2010 [cit. 2010-02-27]. Simple and Intelligent: Application-Specific Vision Sensor. Dostupné z WWW: <<http://www.automation.siemens.com/mcms/industrial-sensors/en/vision-sensor/Pages/default.aspx>>.
- [6] Přehled trhu : inteligentní kamery. *Automatizace* [PDF dokument]. 2009, 5, [cit. 2010-03-21]. Dostupný z WWW: <<http://www.odbornecasopisy.cz/res/pdf/39009.pdf>>.
- [7] Smart kamery pro strojové vidění. *Automatizace* [online]. 2007, 50, 1, [cit. 2010-04-27]. Dostupný z WWW: <<http://www.automatizace.cz/article.php?a=1588>>.
- [8] Inteligentní průmyslové kamery – přehled trhu. *Automatizace* [online]. 2004, 47, 11, [cit. 2010-03-27]. Dostupný z WWW: <<http://www.automatizace.cz/article.php?a=389>>.
- [9] XIE, S.Q., Cheng, D., Wong S., et al. *Three-dimensional object recognition system for enhancing the intelligence of a KUKA robot*. [PDF dokument]. London: Springer-Verlag, 2007. 18 p. Dostupný z: <<http://www.springerlink.com/content/g052112h61264121/>>
- [10] *Hometrica* [PDF dokument]. 2007 [cit. 2010-04-27]. Passive 3D Scanning Systems:. Dostupné z WWW: <<http://www.hometrica.ch/docs/passivesystems.pdf>>.
- [11] *St. Petersburg Conference on Digital Photogrammetry* [online]. St. Petersburg, Russia : SPIE, 1995 [cit. 2010-04-10]. A Comparative Evaluation of the Performance of Passive and Active 3-D Vision Systems, s. . Dostupné z WWW: <[http://scholar.google.cz/scholar?q=A+Comparative+Evaluation+of+the+Performance+of+Passive+and+Active+3-D+Vision+Systems&hl=cs&as\\_sdt=0&as\\_vis=1&oi=scholar](http://scholar.google.cz/scholar?q=A+Comparative+Evaluation+of+the+Performance+of+Passive+and+Active+3-D+Vision+Systems&hl=cs&as_sdt=0&as_vis=1&oi=scholar)>. ISBN NRC39160.

- [12] SICK : *Kamerové systémy : Přehled produktů*, Praha. 2010. 30 s
- [13] KRAJCAR, M. *Robotické vidění s průmyslovými roboty KUKA*. Brno, 2009. 63 s.
- [14] *PPT VISION* [online]. 2009 [cit. 2010-02-28]. IMPACT– Smart Camera Systems for Factory Floor Applications. Dostupné z WWW: <<http://www.pptvision.com/products.html>>.
- [15] *PPT VISION* [online]. 2008 [cit. 2010-04-18]. IMPACT A20. Dostupné z WWW: <[http://www.pptvision.com/product\\_a-series.html](http://www.pptvision.com/product_a-series.html)>.
- [16] *SIEMENS* [online]. 2007 [cit. 2010-04-28]. SIMATIC. Dostupné z WWW: <[http://www.automatyka.siemens.pl/docs/docs\\_pi/SIMATIC\\_VS720.pdf](http://www.automatyka.siemens.pl/docs/docs_pi/SIMATIC_VS720.pdf)>.
- [17] *FESTO* [online]. 2009 [cit. 2010-01-28]. Výrobky a řešení pro vaše úlohy. Dostupné z WWW: <[http://www.festo.com/cms/cs\\_cz/9457.htm](http://www.festo.com/cms/cs_cz/9457.htm)>.
- [18] *FESTO* [online]. 2008 [cit. 2010-04-28]. Zpracování obrazu. Dostupné z WWW: <[http://www.festo.com/cms/cs\\_cz/9511.htm](http://www.festo.com/cms/cs_cz/9511.htm)>.
- [19] *Cognex* [online]. 2010 [cit. 2010-04-28]. Product Overview. Dostupné z WWW: <<http://www.cognex.com/ProductsServices/ProductOverview/default.aspx?id=40&langtype=2057>>.
- [20] *Basler* [online]. 2010 [cit. 2010-04-28]. Basler Components - Machine Vision. Dostupné z WWW: <[http://www.baslerweb.com/beitraege/beitrag\\_en\\_17701.html](http://www.baslerweb.com/beitraege/beitrag_en_17701.html)>.
- [21] *JAI* [online]. 2007 [cit. 2010-04-15]. Camera Solutions. Dostupné z WWW: <<http://www.jai.com/EN/CameraSolutions/Pages/Home.aspx>>.
- [22] DRÁBEK, J. *Sběrnice používané v oblasti programovatelných automatů*. Brno, 2008. 71 s.
- [23] PIRES, J. N. *Industrial Robots Programming: Building Applications for the Factories of the Future*. Springer, 2007. 282 s. ISBN 978-0-387-23325-3
- [24] *ControlTech* [online]. 2010 [cit. 2010-01-20]. DeviceNet. Dostupné z WWW: <<http://www.controltech.cz/products/devicenet.php?pid=2>>.
- [25] *Softyon* [online]. 2000, 2001 [cit. 2010-01-28]. DeviceNet - standardní otevřená síť. Dostupné z WWW: <<http://www.softyon.cz/ab/devicenet.htm>>.
- [26] Síťová architektura Netlinx. *Automatizace* [online]. 2005, 48, 6, [cit. 2010-01-28]. Dostupný z WWW: <<http://www.automatizace.cz/article.php?a=731>>.

- [27] *BECKHOFF* [online]. 2010 [cit. 2010-04-28]. Documentation. Dostupné z WWW: <<http://www.beckhoff.com/>>.
- [28] *Interfacebus* [online]. 1998, 2009 [cit. 2010-02-25]. PC-104 Bus . Dostupné z WWW: <[http://www.interfacebus.com/Design\\_Connector\\_PC104.html](http://www.interfacebus.com/Design_Connector_PC104.html)>.
- [29] SIEMENS. *Spectation Script Programming: Reference Guide* [PDF document]. Nuernburg: 2008. 280 s.
- [30] SIEMENS. *SIMATIC VS72x / Spectation: Manual*. [PDF document]. Nuernburg: 2003. 147 s.
- [31] JAIN, Ramesh; KASTURI, Rangachar; SCHUNCK, Brian G. *Machine Vision*. USA : McGraw-Hill, 1995. 549 s. ISBN 0-07-032018-7.
- [32] SONKA, Milan; HLAVAC, Vaclav; BOYLE, Roger. *Image Processing, Analysis and Machine Vision*. third edition. USA : Thomson Learning, 2008. 829 s. ISBN 978-0-495-08252-1.
- [33] FORSYTH, David A.; PONCE, Jean. *Computer Vision : A Modern Approach*. USA : Pearson Education, 2003. 693 s. ISBN 0-13-191193-7.
- [34] ZEZULKA, František. *Prostředky průmyslové automatizace*. první. Brno : VUTIUUM, 2004. 176 s. ISBN 80-214-2610-1.

## Seznam použitých zkratk

ADC	- Analog to Digital Convertor
CAN	- Controller Area Network
CCD	- Charge-Coupled Device
CMOS	- Complementary Metal-Oxide Semiconductor
CPU	- Central Processor Unit
CVS	- Compact Vision System
DSP	- Digital Signal Processor
FPGA	- Field Programmable Gate Array
HMI	- Human-Machine Interface
I/O Bus	- vstupně výstupní sběrnice
KRL	- Kuka Robot Language
OPC	- Object linking and embedding for Process Automation
PC	- Personal Computer
PCI	- Peripheral Component Interconnect
PLC	- Programmable Logic Controller
RAM	- Random Access Memory
RS232	- rozhraní pro sériovou komunikaci
SoC	- System on Chip
TCP/IP	- Transmission Control Protocol / Internet Protocol
Ucc	- napájecí napětí
USB	- Universal Serial Bus
VGA	- standard pro počítačovou zobrazovací techniku

	Ústav výrobních strojů, systémů a robotiky	Str. 55
	DIPLOMOVÁ PRÁCE	

## Seznam příloh

Příloha 1 – Foreground skript – vybrané části

Příloha 2 – Background skript – vybrané části

Příloha 3 – program pro PLC

Příloha 4 – program pro robot – vybrané části

## Seznam příloh na CD

Příloha 5 – poster k diplomové práci (pdf)

Příloha 6 – video demonstrační úlohy

## Příloha 1 – Foreground skript – vybrané části

```

class AHscript
{
    public void inspect()
    {

//----deklarace----
        String poziceX, poziceY, natoceni, cisloKostky;
        String Hpaket, noObject;
        double cisKostky;

//----skript---
        DebugPrint("zacatek skriptu");

        if (ObjFind1.Result==PASS)
        {
            poziceX=DoubleToString(ObjFind1.ObjectTransformedPoint.X[0],0);
            poziceY=DoubleToString(ObjFind1.ObjectTransformedPoint.Y[0],0);
            natoceni=DoubleToString(ObjFind1.ObjectAngle[0],0);

            cisloKostky=DoubleToString(cisKostky,0);

            Hpaket=poziceX+" "+poziceY+" "+natoceni+" "+cisloKostky+" ";

            DebugPrint("Hpaket : "+Hpaket);

            RegisterWriteString(5000, Hpaket);

            AHscript.String=Hpaket;
        }
        else
        {
            noObject="objektNenalezen";
            RegisterWriteString(5000,noObject);
        }
        RegisterWriteByte(4000,1);

//----konec skriptu----
    }
}

```





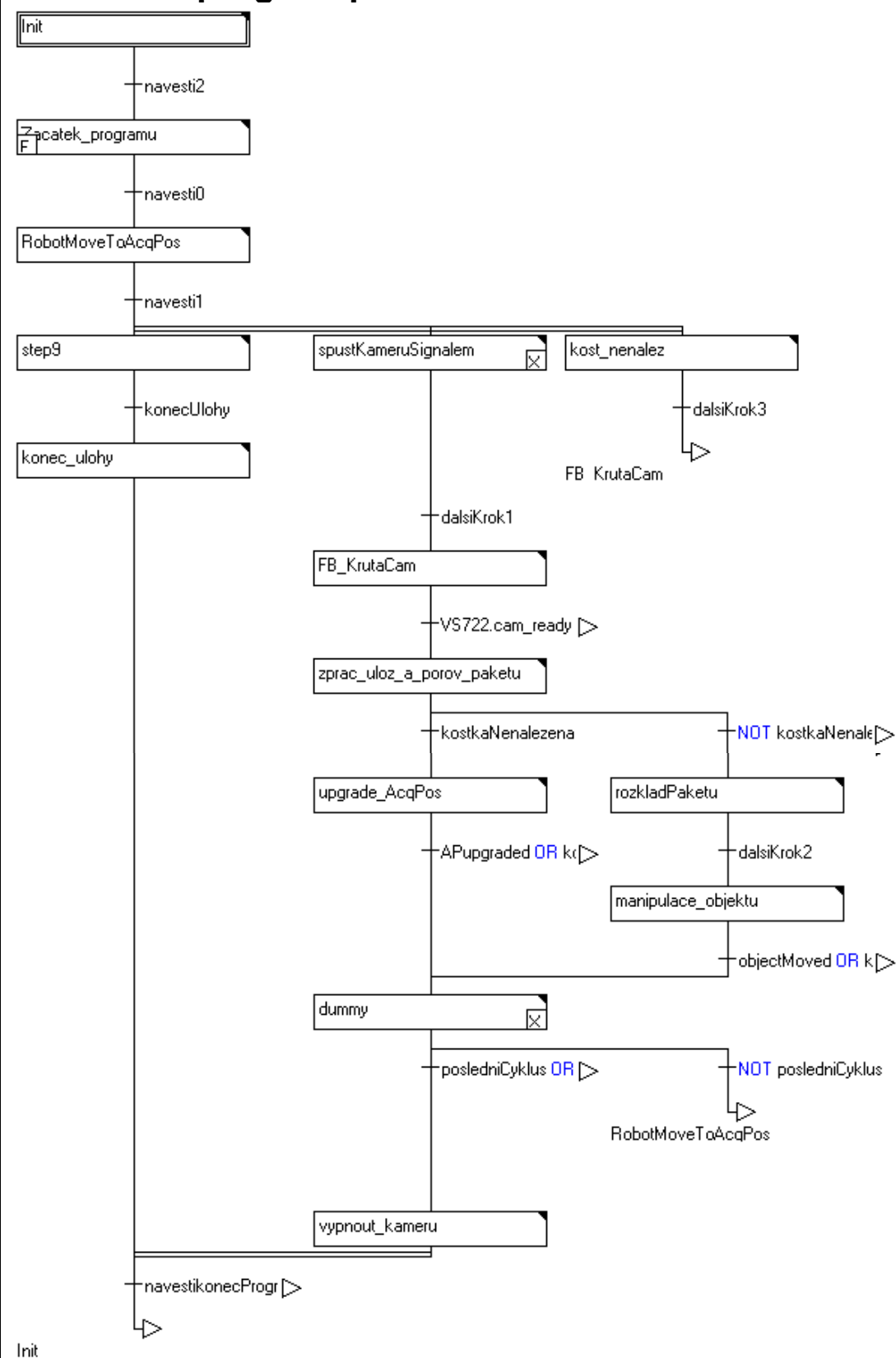
## Příloha 2 – Background skript – vybrané části

```
class BGNewScript
{
    public static Socket inicializace(int AcisloPortu, String
AipAdresa)
    {
        int AcisloPortu=200;
        String AipAdresa="192.168.1.7";
        int AvysledekPripojeni,n=0 ;
        Socket Asock;

        Asock = new Socket();
        AvysledekPripojeni=Asock.Connect(AipAdresa, AcisloPortu);
        DebugPrint("AVysledek pripojeni:"+AvysledekPripojeni);
        while (n!=3)
        {
            if (AvysledekPripojeni<0)
            {
                AvysledekPripojeni=Asock.Connect(AipAdresa,AcisloPortu);
                DebugPrint("AVysledek Pripojeni:"+AvysledekPripojeni);
            }
            n=n+1;
        }
        return Asock;
    }
    public static void main()
    {
        //----deklarace---

        int vysledekPrenosu;
        String paket;
        Socket sock;

        //----zacatek BG skriptu-----
        sock=inicializace(cisloPortu, ipAdresa);
        do
        {
            while (RegisterReadByte(4000)==1)
            {
                paket=RegisterReadString(5000);
                vysledekPrenosu=sock.Send(paket.toByteArray());
                DebugPrint("Vysledek prenosu:"+vysledekPrenosu);
                DebugPrint("paket :"+paket);
                RegisterWriteByte(4000,0);
            }
        }
        while (true);
        sock.Close();
        //----konec BG skriptu
    }
}
```

**Příloha 3 – program pro PLC**

## Příloha 4 – program pro robot – vybrané části

```

DEF KRUTA_01( )

;--- Declaration section ---
EXT BAS (BAS_COMMAND :IN,REAL :IN)
SIGNAL POKUS_05 $IN[9] TO $IN[16]

SIGNAL START ;startovací tlačitko INPUT
SIGNAL ON_THE_AP ;indikace dosazení polohy AP - OUTPUT
SIGNAL GO_OBJECT ;INPUT

;--- Initialization ---
BAS (#INITMOV,0 )

BAS(#ACC_CP,0.5) ; Set Acceleration for linear motions
BAS(#ACC_PTP,30) ; Set Acceleration for axis motions
BAS(#VEL_CP,1) ; Set Speed for linear motions
BAS(#VEL_PTP,15) ; Set Speed for axis motions

BAS(#TOOL,11); Change TOOL
BAS(#BASE,0); Change BASE - NULLFRAME

HOME={A1 0,A2 -90,A3 90,A4 0,A5 30,A6 0}

;--- Main section ---
PTP HOME

LOOP
    WAIT FOR START
    GO_AP()
    PULSE (ON_THE_AP,TRUE,0.2)
    WAIT FOR GO_OBJECT
    IF OBJECT_FOUND THEN GET_OBJECT()
    ELSE UPDATE_SECTOR()
    ENDIF
ENDLOOP

END

DEF GO_AP()
    PTP AP
END

DEF GET_OBJECT()

```



## DIPLOMOVÁ PRÁCE

```
PTP OBJECT_POS
GRASP ( )
PTP STORAGE_POS
UNGRASP ( )
END

EF MAKE_POS ( )
  INT x,y,z,a,b,c,dx,dy
  x = 0
  y = 0
  z = 100
  a = 0
  b = 0
  c = 180

  dx = 60
  dy = 120

  FOR ROW = 1 TO 2
    FOR COLUMN = 1 TO 4
      SECTOR_BASES[COLUMN,ROW].x = x
      SECTOR_BASES[COLUMN,ROW].y = y
      SECTOR_BASES[COLUMN,ROW].z = z
      SECTOR_BASES[COLUMN,ROW].a = a
      SECTOR_BASES[COLUMN,ROW].b = b
      SECTOR_BASES[COLUMN,ROW].c = c
      y = y + dy
    ENDFOR
    x = x + dx
    y = 0
  ENDFOR
END

DEF GRASP ( )
  PTP {Z 5}
  VACUM ON
END

DEF UNGRASP ( )
  VACUM OFF
END
```